

Advant[®] OCS

AdvaInform[®] Reports

User's Guide



Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaInform and AdvaBuild are registered trademarks of ABB Industrial Systems Inc.

MOD 300, Batch 300, and Taylor are Trademarks of ABB Industrial Systems Inc.

ORACLE, SQL*Plus are registered trademarks of Oracle Corporation.

Adobe, Acrobat, and PostScript are registered trademarks of Adobe Systems Incorporated.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrial Systems Inc.. ABB Industrial Systems Inc. assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrial Systems Inc. be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrial Systems Inc. be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrial Systems Inc.'s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © 1997 ABB Industrial Systems Inc. All rights reserved.

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1	General Information.....	1-1
1.2	Equipment Requirements.....	1-2
1.3	Manual Organization	1-2
1.3.1	Differences Between Master and MOD 300.....	1-4
1.4	Conventions	1-5
1.5	Related Documentation.....	1-6
1.6	Release History	1-7
1.7	Terminology.....	1-8
1.8	Product Overview	1-10
1.8.1	Data Access for AdvaInform Reports.....	1-11
1.8.1.1	Standard Views for Advant OCS Objects	1-12
1.8.1.2	SQL Queries.....	1-13
1.8.2	Access to AdvaInform Reports.....	1-13
1.8.3	Report Configuration	1-14
1.8.3.1	The Report Template.....	1-14
1.8.3.2	Report Object Instance	1-17
1.8.4	Manual Execution	1-21
1.8.5	Report Status.....	1-22
1.8.6	View Report Log.....	1-23
1.8.6.1	Viewing and Printing Reports Stored in Report Logs....	1-23
1.8.6.2	Modifying Character Mode Reports Stored in Report Logs.....	1-23
1.8.7	Report Execution	1-24

Chapter 2 - Installation

2.1	Site Planning Environment	2-1
2.1.1	Guidelines for planning installation of Advant Station	2-1
2.2	Setup	2-1
2.3	Start-up Procedures.....	2-1
2.4	Product Verification	2-1

Chapter 3 - Configuration and Application Building

3.1	Design Considerations	3-1
3.1.1	Printer Considerations.....	3-1
3.1.2	AdvaInform History Considerations.....	3-1
3.1.3	User API Considerations	3-1
3.1.4	Batch/TCL Considerations (MOD 300 Only)	3-2

CONTENTS (continued)

3.1.5	Building the AdvaInform SQL*Connect Location Table	3-2
3.1.6	Building Report Logs.....	3-2
3.1.7	Report Output Mode	3-3
3.1.8	Making Reports Accessible on the Web	3-4
3.2	Capacity and Performance	3-4
3.3	Application Start-up	3-4
3.3.1	Starting Reports on Advant Station w/AdvaCommand (MOD 300 Only).....	3-4
3.3.2	Starting Reports on Advant Station w/AdvaInform.....	3-4
3.3.3	How to Configure the Report Output for Character Mode	3-5
3.4	Tutorial	3-9
3.4.1	Open the Report Configuration Window	3-10
3.4.2	Build the Report Template	3-11
3.4.2.1	Open the Object Navigator	3-11
3.4.2.2	Select the Output Mode for the Report.....	3-11
3.4.2.3	Create a New Report Template.....	3-12
3.4.3	Create a New Report Object Instance	3-24
3.4.3.1	Assign the Report Template to the Report Object Instance.....	3-25
3.4.3.2	Enable Scheduling Modes	3-25
3.4.3.3	Configure the Parameter List (Execution Parameters)...	3-26
3.4.3.4	Schedule the Report for Execution.....	3-30
3.5	Application Procedures	3-39
3.5.1	How to Open the Report Configuration Window	3-39
3.5.2	How to Build Report Templates with Oracle Reports	3-39
3.5.2.1	How to Open an Existing Report	3-40
3.5.2.2	How to Make Data Available to Other Reports (Encapsulate Data).....	3-41
3.5.2.3	How to Use Execution Parameters in a Query	3-45
3.5.3	How to Create a Report Object Instance.....	3-45
3.5.4	How to Back Up and Restore Report Templates	3-45
3.5.5	How to Integrate Graphic Charts in a Report.....	3-46
3.5.5.1	Tutorial 1 - Basic Bar Chart	3-46
3.5.5.2	Tutorial 2 - Integrating Graphics with Queries that Use Functions	3-54
3.5.6	How to Send Report Output to a File.....	3-58
3.5.7	How to Print a Report	3-61
3.5.7.1	Printing in the Landscape Mode.....	3-61
3.5.8	How to Store Report Output in History	3-61

CONTENTS (continued)

3.6	Configuration/Application Building Menus	3-62
3.6.1	Reports Menu.....	3-62
3.6.1.1	Report Configuration	3-62
3.6.1.2	Manual Execution	3-62
3.6.1.3	Report Status	3-62
3.6.1.4	View Logs	3-62
3.6.2	Report Configuration Window	3-63
3.6.2.1	Parameter List Configuration Dialog.....	3-65
3.6.2.2	File Selection Dialog Box	3-68
3.6.2.3	Parameter Edit Dialog Box	3-69
3.6.2.4	Execution Type Select Dialog Box (Master Only)	3-70
3.6.2.5	Execution Configuration Dialog Box	3-70
3.6.2.6	Month Configuration Dialog Box	3-74
3.6.2.7	Day Configuration Dialog Box	3-75
3.6.2.8	Hour Configuration Dialog Box	3-77
3.6.2.9	Minute Configuration Dialog Box	3-78
3.6.2.10	Second Configuration Dialog Box	3-79
3.6.2.11	Defining Event Conditions (Master Only).....	3-80
3.6.2.12	Report Configuration Window - File Menu.....	3-82
3.6.2.13	Report Configuration Window - Window Menu	3-83
3.7	How to Write Queries for Reports	3-84
3.7.1	Relational Database Structure.....	3-84
3.7.2	SQL Query Structure	3-84
3.7.3	Views of Advant OCS Object Types	3-86
3.7.4	How to See Which Attributes are Available in an Object Type	3-87
3.7.5	Queries for Process Data (MOD 300 Examples).....	3-87
3.7.5.1	Continuous Loop with no Controller FCMs	3-88
3.7.5.2	FCM Attributes in Continuous Loop	3-88
3.7.5.3	Auto/Manual Controller Loop	3-88
3.7.5.4	PID Controller Loop	3-89
3.7.5.5	Device Loop	3-89
3.7.6	Queries for Numeric History Data.....	3-90
3.7.6.1	When to Use Log Name or Access Name.....	3-92
3.7.6.2	Basic Query for Numeric Data.....	3-93
3.7.6.3	Query for Numeric Data from a Secondary Log.....	3-93
3.7.6.4	Query for a Specific Time Range.....	3-94
3.7.6.5	Query for Interpolated Data	3-94
3.7.6.6	Query Based on Calculation Algorithm	3-95

CONTENTS (continued)

3.7.6.7	Query for a Specific Instance in Time	3-95
3.7.6.8	Criteria for Seamless Retrieval.....	3-96
3.7.7	Queries for Messages	3-98
3.7.7.1	Basic Query for Historical Messages	3-99
3.7.7.2	Query for Historical Messages in Time Range	3-99
3.7.7.3	Query for a Specific Type of Historical Message	3-99
3.7.7.4	Combine Message Type and Time Range	3-99
3.7.7.5	Query for a Message with a Specified Text String.....	3-100
3.7.8	How to Read Basic and User Objects	3-101
3.7.9	How to Join Data from Two Tables in a Single Query	3-101

Chapter 4 - Runtime Operation

4.1	Product Operation	4-1
4.2	Operating Overview	4-1
4.2.1	Operator Functions Supported	4-1
4.2.2	Operator Interface	4-2
4.3	Runtime Tutorial	4-2
4.4	Operating Instructions	4-2
4.4.1	How to View the Report Status	4-2
4.4.2	How to Activate or Deactivate Scheduling for a Report.....	4-3
4.4.3	How to Enable or Disable Report Scheduling Modes	4-3
4.4.4	Manual Execution	4-6
4.4.5	How to Manipulate Reports in Historical Storage	4-9
4.4.5.1	How to View a List of Report Logs.....	4-9
4.4.5.2	How to View a List of Reports in a Report Log.....	4-11
4.4.5.3	How to View Reports	4-12
4.4.5.4	How to Print Reports	4-12
4.4.5.5	How to Modify Reports.....	4-13
4.4.5.6	How to Delete a Report from a Report Log	4-14
4.4.5.7	How to Delete Many Reports from a Report Log	4-14
4.4.5.8	How to View a List of Restored Reports in a Report Log.....	4-15
4.4.5.9	How to Search for Reports in Historical Storage	4-16
4.4.5.10	Archiving Finished Reports.....	4-16
4.4.5.11	How to Copy Report Logs to a File	4-17
4.4.6	How to Retrieve Encapsulated data	4-18

CONTENTS (continued)

4.5	Runtime Operation Menus.....	4-19
4.5.1	Manual Report Execution Window Menus.....	4-19
4.5.1.1	Manual Report Execution File Menu.....	4-19
4.5.1.2	Manual Report Execution Window Menu	4-19
4.5.2	View Logs Menus	4-20
4.5.2.1	View Logs File Menu.....	4-20
4.5.2.2	View Logs Action Menu	4-20
4.5.2.3	View Logs Window Menu	4-21
4.5.3	Reports List Menus.....	4-21
4.5.3.1	Reports List File Menu	4-21
4.5.3.2	Reports List Action Menu.....	4-22
4.5.3.3	Reports List Window Menu	4-22

Chapter 5 - Maintenance

5.1	Preventive Maintenance.....	5-1
5.2	Fault Finding and Repair	5-1
5.3	Preventive Maintenance.....	5-1
5.3.1	Maintainability.....	5-1
5.3.2	The HP-UX kernel.....	5-1
5.4	Hardware Indicators.....	5-1
5.5	Error Messages	5-2
5.5.1	Error Messages for MOD 300 Applications	5-2
5.5.1.1	Where to Read Error Messages.....	5-2
5.5.1.2	Error Message Format.....	5-3
5.5.1.3	Reports Diagnostic Messages	5-3
5.5.2	Error Messages for Master Applications	5-13
5.5.2.1	Messages Box	5-13
5.5.2.2	Reports Diagnostic Messages	5-14
5.5.2.3	Fatal error messages.....	5-14
5.5.2.4	Category Error.....	5-15
5.6	Frequently Asked Questions and Problems.....	5-18

Appendix A - SQL Statement Reference for AdvaInform Reports

A.1	General.....	A-1
A.2	SELECT.....	A-1
A.2.1	Purpose.....	A-1
A.2.2	Prerequisites.....	A-1
A.2.3	Syntax	A-1
A.2.4	Keywords and Parameters	A-2

CONTENTS (continued)

A.2.5	Usage Notes	A-2
A.2.6	GROUP BY, HAVING	A-3
A.2.7	Set Operators	A-3
A.2.8	ORDER BY	A-3
A.2.9	Sequences	A-3
A.2.10	Subquery	A-3
A.2.11	Examples	A-3
A.2.12	Related Topics	A-4
A.3	SELECT STATEMENT CLAUSES	A-4
A.3.1	FROM clause	A-4
A.3.1.1	Purpose	A-4
A.3.1.2	Prerequisites	A-4
A.3.1.3	Syntax	A-4
A.3.1.4	Keywords and parameters	A-4
A.3.1.5	Usage Notes	A-5
A.3.1.6	Examples	A-5
A.3.1.7	Related Topics	A-5
A.3.2	GROUP BY and HAVING clauses	A-5
A.3.2.1	Purpose	A-5
A.3.2.2	Prerequisites	A-6
A.3.2.3	Syntax	A-6
A.3.2.4	Keywords and Parameters	A-6
A.3.2.5	Usage Notes	A-6
A.3.2.6	HAVING clause	A-6
A.3.2.7	Examples	A-7
A.3.3	WHERE clause	A-8
A.3.3.1	Purpose	A-8
A.3.3.2	Prerequisites	A-8
A.3.3.3	Syntax	A-8
A.3.3.4	Keywords and Parameters	A-8
A.3.3.5	Usage Notes	A-8
A.3.3.6	Examples	A-9
A.3.3.7	Using the TO_DATE Function in a Condition	A-9
A.3.4	ORDER BY Clause	A-11
A.3.4.1	Purpose	A-11
A.3.4.2	Prerequisites	A-11
A.3.4.3	Syntax	A-11
A.3.4.4	Keywords and Parameters	A-11

CONTENTS (continued)

A.3.4.5	Expr(ession)	A-12
A.3.4.6	Usage Notes	A-12
A.3.4.7	Example	A-12
A.3.4.8	Related Topic	A-13
A.3.5	Simple Join	A-13
A.3.5.1	Purpose	A-13
A.3.5.2	Prerequisites	A-13
A.3.5.3	Syntax	A-13
A.3.5.4	Usage Notes	A-13
A.3.5.5	Cartesian product	A-14
A.3.5.6	Examples	A-14
A.3.5.7	Related Topics	A-15
A.3.6	Outer Join	A-15
A.3.6.1	Purpose	A-15
A.3.6.2	Prerequisites	A-15
A.3.6.3	Syntax	A-15
A.3.6.4	Usage Notes	A-15
A.3.6.5	Example	A-16
A.3.6.6	Related Topics	A-17

INDEX

Chapter 1 Introduction

1.1 General Information

This book describes how to use AdvaInform® Reports software to configure, schedule, and execute reports. With AdvaInform Reports (referred to in this book as Reports) you can execute process reports which contain both current and historical information. Reports can be displayed on the screen, sent to printers, or stored as a file (ASCII, postscript, pdf, html). They can also be stored in AdvaInform History.

This book describes AdvaInform Reports for both Advant OCS with Master software and with MOD 300 software. There are some differences in the application and operation of the Reports package between Master and MOD 300. These are mainly related to differences in AdvaInform object types for MOD 300 and Master, AdvaInform software options for MOD 300 and Master, and the types of Advant® Stations that AdvaInform Reports can run on. Wherever such differences occur in the book they are marked with a parenthesis (Master) or (MOD 300). A summary of differences is given in [Section 1.3.1, Differences Between Master and MOD 300](#).

Reports is an optional package runs on an Advant® Station with AdvaInform software. In systems with MOD 300 software, the Reports package can also run on an Advant Station with AdvaCommand® software. The operation of the Reports package is the same whether you run it on an Advant Station with AdvaInform or AdvaCommand software. The instructions in this book are applicable for both cases. The only difference is how you access Reports. Details are provided in [Section 3.3, Application Start-up](#).

Reports can be made available to PC clients equipped with web browsers by storing the reports in pdf or html format in a directory that is accessible from a web server.

Reports supports four basic functions:

- Report Configuration which involves:
 - building a report template via Oracle® Reports. Oracle Reports is embedded in the AdvaInform Reports software package and must be used as such. The report template defines the content and layout of the report.
 - mapping execution, routing, and scheduling parameters to the report template. This is done via the AdvaInform Reports configuration tool.
- Report Execution. Reports can be executed in the following ways:
 - cyclically according to a specified schedule
 - manually on a one-time basis
 - triggered by a user API program
 - triggered by a REPORT statement in a TCL program (MOD 300 only)
 - event-driven (Master only).
- Viewing report status
- Viewing and modifying reports stored in AdvaInform History.

This book is not intended to be the sole source of instruction for Reports. It is recommended that you attend the appropriate training programs available from ABB Industrial Systems.

1.2 Equipment Requirements

The hardware platform for Reports is the Advant Station 500 Series. The station is equipped with the hardware and software required to support the software packages that you order. The memory provided to support basic functions in these stations, and the memory requirements for various options are described in the *Advant Station Information Management Station User's Guide*, and the *Advant Station Operator Station User's Guide*.

1.3 Manual Organization

This book is organized in chapters and sections as shown in [Figure 1-1](#). This is the general organization for all Advant product user's guides. Some chapters and sections may not apply to AdvaInform Reports.

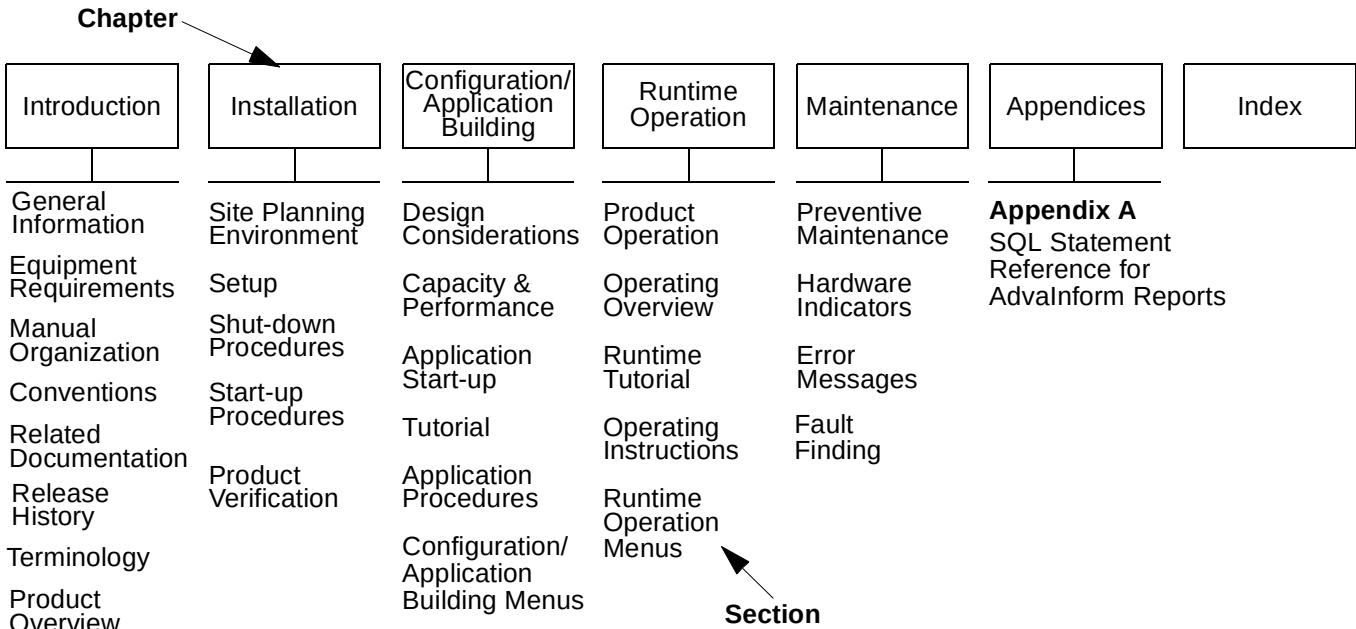


Figure 1-1. Organization of Advant Technology Documentation

How to Use this Book

This book covers both report configuration ([Chapter 3, Configuration and Application Building](#)) and runtime operation ([Chapter 4, Runtime Operation](#)). Read [Section 1.8, Product Overview](#) to get a general understanding of the configuration and runtime functions supported by AdvaInform Reports.

[Chapter 3, Configuration and Application Building](#) is organized so you can learn how to use the Report Configurator without having to read this book from cover to cover. First, refer to the tutorial in Section 3.4 to get some quick “hands-on” experience. When you are finished with the tutorial, you should have a good understanding of the operation of most functions. If you need further information regarding a procedure that is not covered in the tutorial, refer to [Section 3.5, Application Procedures](#). This section provides detailed instructions for basic report building procedures. [Section 3.6, Configuration/Application Building Menus](#) briefly describes the purpose and operation of each menu item.

[Chapter 4, Runtime Operation](#) provides instructions and reference information related to runtime operation such as how to execute reports manually, view logs and so on.

1.3.1 Differences Between Master and MOD 300

The following is a brief summary of the main differences in the application and operation of AdvaInform Reports for Master and MOD 300.

Reports on an Advant Station with AdvaCommand Software

In systems with MOD 300 software, AdvaInform Reports can run on an Advant Station with either AdvaInform software or AdvaCommand software. In systems with Master software, AdvaInform Reports runs on an Advant Station with AdvaInform software. All references in this book to Reports on an Advant Station with AdvaCommand software are for MOD 300 applications only.

Object Types

While some AdvaInform object types are common for both Master and MOD 300, there are also some object types that are unique to Master or MOD 300. A set of pre-defined Oracle views are available to facilitate data retrieval for these object types. Some views are common for Master and MOD 300, and some are unique. These views are described in [Section 1.8.1.1, Standard Views for Advant OCS Objects](#). Refer to the applicable Object Type Reference Manual for details. There is a separate manual for MOD 300 object types and Master object types.

MEASURE and VALUE Object Attributes

One common object attribute for MOD 300 report applications is the MEASURE attribute. The equivalent attribute for Master applications is VALUE. These attributes are used in many of the examples in this book. In some cases, two examples are provided showing both MEASURE and VALUE, and in other cases just one attribute or the other is shown. In all cases, the examples can be made applicable for both Master and MOD 300 by simply substituting the correct attribute.

Event and Alarm Handling

Event and alarm handling differs for MOD 300 and Master. All examples and other references to event handling in this book are applicable for Master only.

TCL and CCF

TCL and CCF are software packages for MOD 300 applications. All examples and references for TCL and CCF are only applicable for MOD 300.

AdvaInform Options

Different AdvaInform options are available for MOD 300 and Master. For this reason, there may be different menu items available under the **AdvaInform** menu in the IMS menu bar for MOD 300 and Master.

1.4 Conventions

The following conventions are used throughout this manual for the presentation of material:

- The words in proper names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- The names of keyboard keys are boldfaced.
 - Initial capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, the **Return** key or **<Return>**.
 - Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
 - In descriptions, the word “key” is used in conjunction with its name. For example, pressing the **Return** key initiates the process.
 - In tables and procedures, angle brackets replace the word “key.” For example, press **<Return>**.
 - A dash is used to indicate that you must simultaneously press several keys. For example, **<Ctrl-C>** indicates that you must hold down the **Control** key while pressing the **C** key.
 - The phrase “press and release” is used to indicate that you sequentially press several keys. For example, to close a window, press and release **<Alt>**, and then **<F4>**. This indicates that you press and release the **Alt** key, and then you press and release the **F4** function key.
- The names of push and toggle buttons are boldfaced.
 - Initial capital letters are used for the name of a push or toggle button labeled on the display. For example, the **OK** button.
 - In descriptions, tables and procedures, the word “button” is used in conjunction with its name. For example, pressing the **OK** button initiates the process.
- The names of menus and menu items are boldfaced.
 - Initial capital letters are used for the name of menus listed in the menu bar. For example, the **File** menu.
 - Initial capital letters are used for the name of menu items. For example, to open a new window, choose **New** from the **File** menu.
- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

Entered value is not valid. The value must be 0-30.

In a tutorial, you may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure:

TIC132

1.5 Related Documentation

Documentation related to AdvaInform Reports is shown in [Table 1-1](#).

Table 1-1. Related Documentation

Category	Title	Description
Hardware/Station-Level	Advant Station 5nn Hardware User's Guide	There are several Advant Station Hardware user's guides, one for each version of the Advant Station. The <i>Advant Station Hardware User's Guide</i> covers the installation, and maintenance of the hardware for an Advant Station, whether it is equipped with AdvaInform software or AdvaCommand software.
	Advant Station 500 Series IMS User's Guide & Advant Station Operator Station User's Guide	The <i>Advant Station 500 Series IMS User's Guide</i> and <i>Advant Station Operator Station User's Guide</i> provide instructions for system administration functions including software installation and backup and restore.
AdvaInform Software	AdvaInform Basic Functions User's Guide	The <i>AdvaInform Basic Functions User's Guide</i> describes the basic functions that support all AdvaInform software packages.
	AdvaInform Object Types Reference Manual	The <i>AdvaInform Object Types Reference Manual</i> provides the names of data base tables and attributes required for accessing information for the reports. There are separate reference manuals for MOD 300 and Master.
	AdvaInform History User's Guide	The <i>AdvaInform History User's Guide</i> describes how to build report logs for storing reports in AdvaInform History.
Oracle Reports 2.5 Documentation	Oracle Developer/2000 Reports 2.5 Building Reports Manual	The Oracle Developer/2000 documentation provides information on Oracle Reports not covered in this book. This documentation is available from ABB. If you are going to access report information from Oracle® data bases in host computers on the plant network, you need the SQL*Net® for TCP/IP installed in the host computers. Refer to the SQL*Net for TCP/IP documentation for your host computer.
	Oracle Developer/2000 Reports 2.5 Reference Manual	
	Oracle Developer/2000 Graphics 2.5 Developers Manual	
	Oracle Developer/2000 Graphics 2.5 Reference Manual	

1.6 Release History

This book describes AdvaInform Reports release 2.2. The following is a summary of the releases for AdvaInform Reports:

- | | |
|----------|---|
| Rel. 1.0 | Initial release of the AdvaInform Reports software for the Advant Station 500 Series. |
| Rel. 1.1 | General enhancements and TCL Access (MOD 300 only) |
| Rel. 1.2 | General enhancements, multiple delete functionality, and verification of directory on Save function. |
| Rel 2.0 | Oracle Reports version 2.5. This version provides a better user interface, and supports import of bitmaps into reports. |
| Rel. 2.2 | <p>Reports can be stored as PDF or HTML files.</p> <p>Reports that are stored in any of the four formats (Character, Graphic, PDF, HTML) in Historical Report logs can be moved to files in /var/opt/advant/Reports/ via file specification dialog.</p> <p>Multiple instances of a report can be run and stored in different files using the same parameter values in successive executions. This is done via the reportprefix and reportsuffix parameters. These parameters cause subsequent executions of a report to be placed in a directory beneath the default report directory: \$reportprefix with a name \$reportprefix_\$date\$suffix. The maximum number of instances that can be stored defaults to 50, and can be adjusted via the reportdepth parameter.</p> <p>The batchprefix, batchsuffix and batchdepth parameters provide equivalent functionality for batch reports.</p> <p>For postscript and text report printer output: adding parameter "landscapemode" and assigning it to any value will cause the lp command to have compressed landscape mode options passed.</p> |

1.7 Terminology

This section gives brief definitions of terms related to Reports.

Relational Database (RDB)

A relational database organizes information into tables on the basis of similar characteristics. The data in an RDB table is organized in columns and rows in much the same way as data in text book tables.

Structured Query Language (SQL)

SQL is a language for manipulating relational databases. It consists of commands to store, retrieve, maintain, and regulate the database.

AdvaInform SQL*Connect

AdvaInform SQL*Connect provides a method for SQL to read information from Advant OCS objects.

AdvaInform User API (Application Programmer's Interface)

The AdvaInform User API can be used by the customer programmers for building applications.

Report Template

A template defines the content (data base information in the report) and layout (presentation) of the report.

Execution Parameter

A place holder for a variable in a report template. An actual value is substituted for the execution parameter when the report is executed. The values to be substituted for execution parameters are defined in parameter lists via the Report Configurator. This way you can insert current information into an instance of the report. The batch or object name is often an execution parameter.

Report Object Instance

A report object instance contains the specifications required to execute a finished report. A report object instance is produced whenever you map parameters and conditions to the report template using the Report Configurator. For example, you may have a format for producing a batch end report. When you map the parameters of batch 15 to it, you have a report object instance which can be executed to produce a report for batch 15. You can similarly map parameters for batch 16 to the format to produce a report for batch 16.

Encapsulated Data

Encapsulated data is data from a finished report that can be inserted into another report.

Report Execution

Report execution is the act of creating a finished report from a report object instance.

Finished Report

The results of a report execution.

Report Presentation

Report presentation is the viewing of a finished report on the screen or printing of the report on a printer.

Report Logs

Report Logs are the AdvaInform History objects within the historical database that receive finished reports and the encapsulated data for storage.

Restored Report Log

Restored report logs are report logs that have been restored from archive media through the AdvaInform History archive tool.

1.8 Product Overview

AdvaInform Reports organizes, presents, prints, and stores data related to the automation process. [Figure 1-2](#) shows the AdvaInform Reports components.

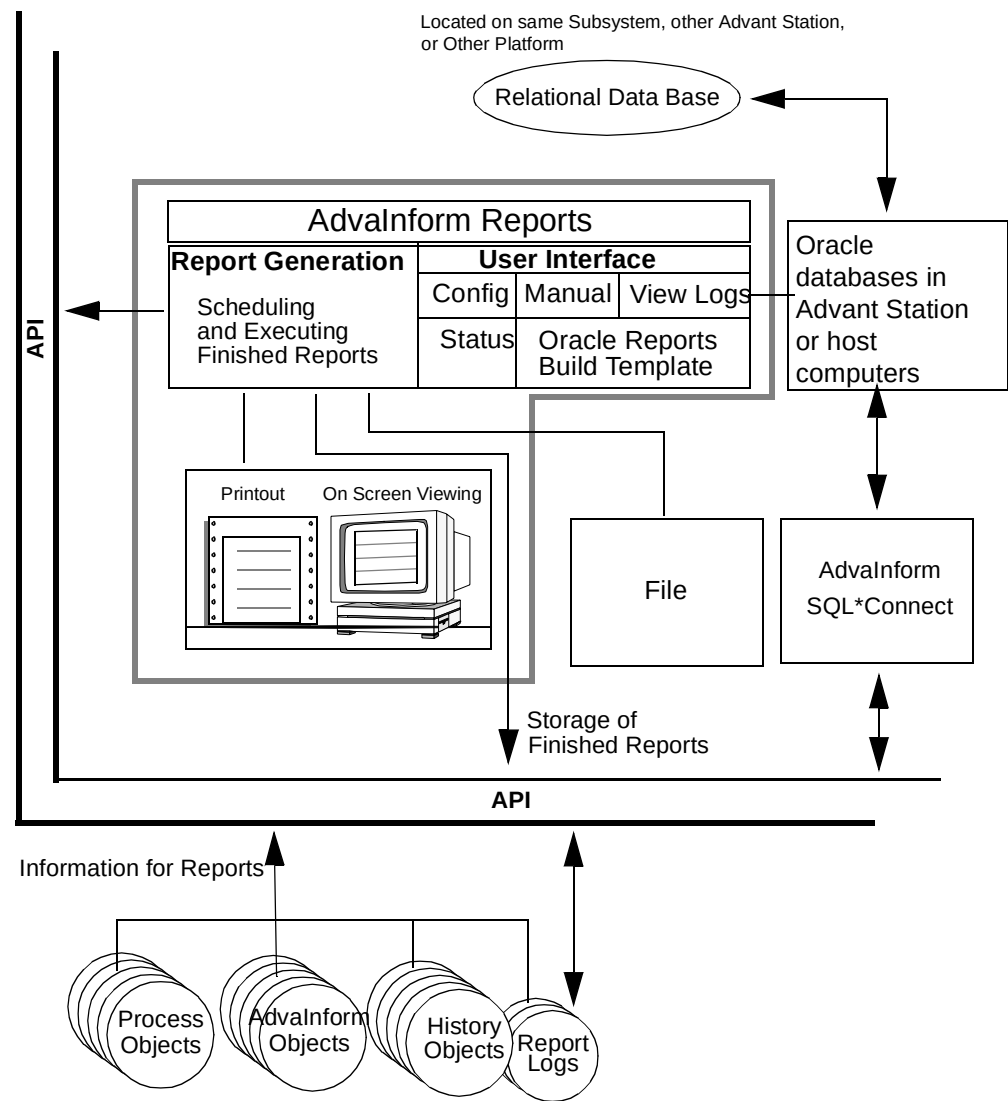


Figure 1-2. AdvaInform Reports Components

1.8.1 Data Access for AdvaInform Reports

The reports you build with AdvaInform Reports retrieve data from Advant OCS objects via SQL through AdvaInform SQL*Connect, [Figure 1-3](#). The data are retrieved by SQL queries that read data from tables and views of relational data bases. Data from external Oracle databases are retrieved via SQL over SQL*Net. Data from other databases can be retrieved via connect products from Oracle.

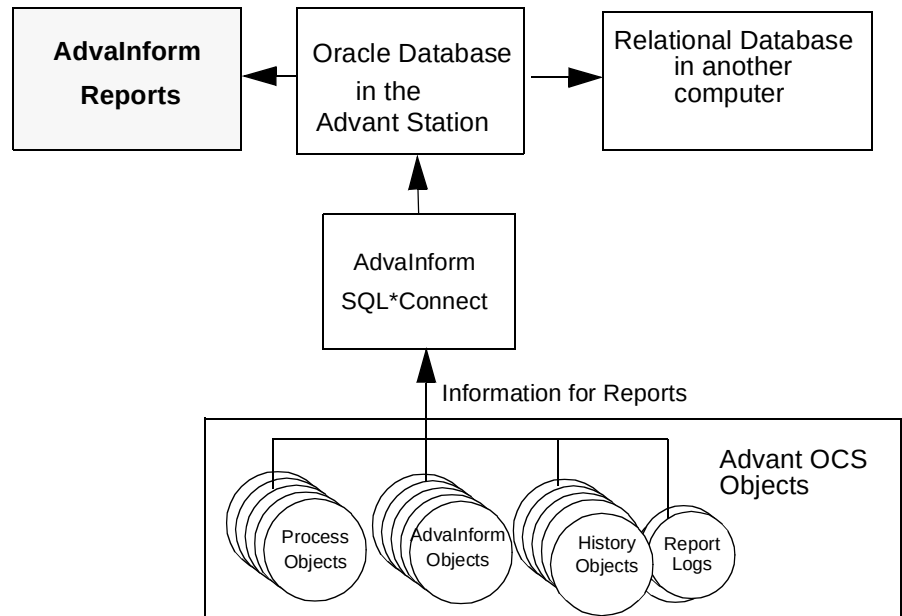


Figure 1-3. Data Access for AdvaInform Reports

AdvaInform SQL*Connect makes the Advant OCS objects appear as relational tables. This allows reports to access the following types of data:

- Process information such as measured values from loops (Process objects)
- Information from AdvaInform History objects such as:
 - Process variable values
 - Alarm and event messages
 - Batch execution information
- Information from AdvaInform Basic objects or User objects.

In addition, data can also be retrieved from:

- Oracle databases in the Advant Station. These databases are usually related to customer applications.
- Oracle data bases on host computers on the plant network. This requires SQL*Net TCP/IP running in the host computer. Also, Oracle must be installed on the host computer such that it is accessible by other Oracle nodes (not stand-alone).

1.8.1.1 Standard Views for Advant OCS Objects

Views are custom representations of Oracle tables that facilitate data access. Views are created by grouping columns from one or more tables. Once the view is created, you can read data from the view rather than the tables.

Some pre-defined views are provided for you to facilitate data access from Advant OCS objects. These views are described in [Table 1-2](#). The *AdvaInform Object Type Reference Manual* contains complete listings of these views.

Table 1-2. Pre-defined Tables and Views for Advant OCS Objects

System	Table Name	Purpose
MOD 300	CCF_CONTIN_LOOP	Loop values from continuous loops which have no Auto/Manual or PID Controller FCMs
	CCF_CNTRL_LOOP	Loop and Auto Manual Controller FCM attributes from continuous loops which have Auto/Manual FCMs.
	CCF_PID_LOOP	Loop and PID FCM attributes from continuous loops which have PID FCMs but do not use adaptive reset or adaptive gain.
	CCF_1ADAPT_LOOP	Loop and PID FCM attributes from continuous loops which have PID FCMs and use either adaptive reset or adaptive gain.
	CCF_2ADAPT_LOOP	Loop and PID FCM attributes from continuous loops which have PID FCMs and use both adaptive reset and adaptive gain
	FCM object types	Each FCM type has an object type associated with it. For example, math_fcm, ainp_fcm, and so on.
	CCF_DEVICE_LOOP	Device loop attributes
	DATA_FCM, VA_STRING	Reading values from AdvaInform Basic Objects from either the local or remote nodes.
Master	AI, AO, DI, DO, TEXT, DAT	Reading values from AdvaInform Basic Objects from either the local or remote nodes. AI is used to read AI objects, AO for AO objects, and so on.
Common (MOD 300 & Master)	NUMLOGVAL	Historical values from numeric historical logs
	HS_MSG	For MOD 300, this contains messages from all historical message logs. For Master, to get a more concise view of EAMSGLOG, use BCHS_MESSAGES.
	ENCAP_DATA	Used by AdvaInform Report users to put data from an executed report into a new report
	IMS_TABLES	Contains names of tables to which the current AdvaInform user has privileges.

1.8.1.2 SQL Queries

The queries for AdvaInform Reports follow SQL language conventions. A query consists of a short group of statements that starts with a Select statement followed by other clauses.

A Select statement (query):

- identifies the database attributes to be returned.
- identifies the tables or views that contain the attributes.
- gives conditions that determine which instances in the table or view are to provide data.
- specifies the order to present the returned data.

All features of a Select statement are supported except the For Update clause. As an example, the query in [Figure 1-4](#) says:

Retrieve the entryvalue attribute from the NUMLOGVAL view for all instances where the NAME = SC5_FC101. No order is specified for the returned data.

```
SELECT ENTRYVALUE FROM NUMLOGVAL
WHERE NAME = SC5_FC101
```

Figure 1-4. Example Query

1.8.2 Access to AdvaInform Reports

On the Advant Station with AdvaInform software, you access AdvaInform Reports from the IMS Menu, [Figure 1-5](#). On the Advant Station with AdvaCommand software, you access it from display menus (when the station environment is configured to support that functionality).

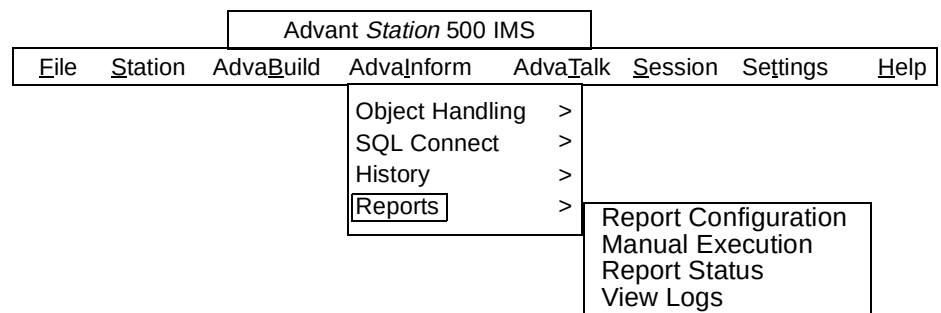


Figure 1-5. Accessing AdvaInform Reports from IMS Menu

The functionality provided by the four **Reports** menu items is described in the following sections.

1.8.3 Report Configuration

Report configuration is a two-part process:

- First you build a report template.

Report templates are built using Oracle Reports. The Oracle Reports is a product of the Oracle corporation, However, ABB has added some functionality to it and has embedded it within AdvaInform Reports. Access to Oracle Reports is provided by the Report Configuration window, [Figure 1-6](#).

- Then you create a report object instance.

You create the report object instance by mapping execution and scheduling parameters to the report template. This is done via the Report Configuration window.

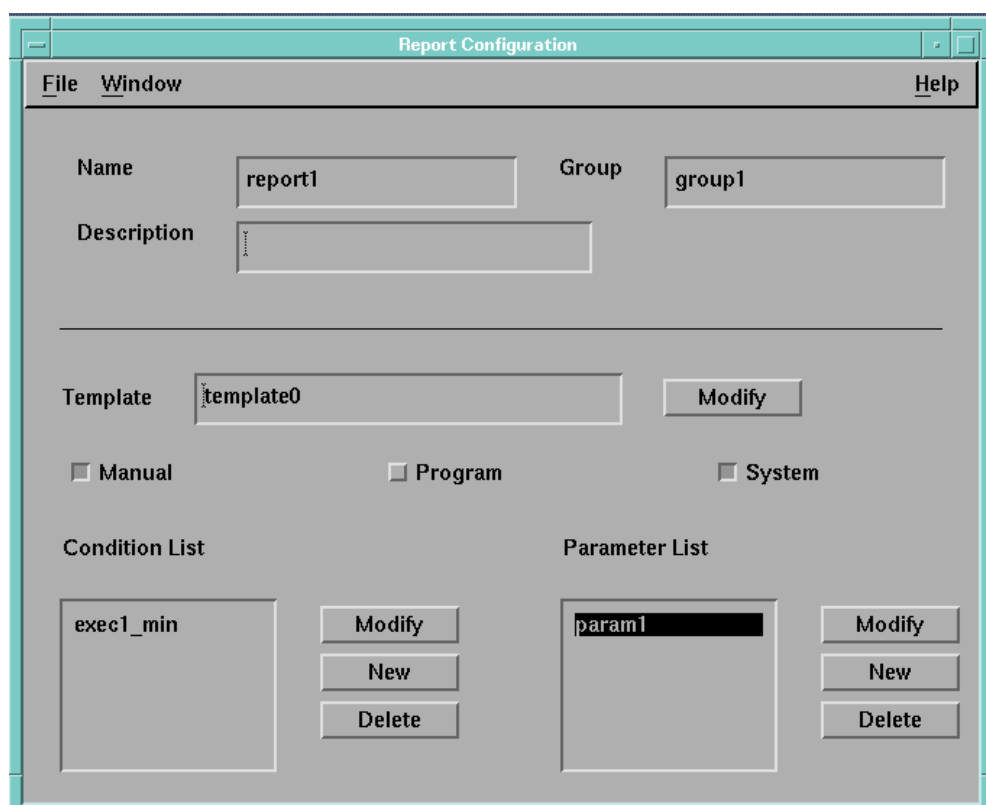


Figure 1-6. Report Configuration Window

1.8.3.1 The Report Template

The tool for building report templates is Oracle Reports which is based on SQL (Structured Query Language). You build report templates by defining one or more SQL queries that retrieve data from relational databases. In addition to the SQL queries, the template defines the report layout (how data are presented and labeled).

You invoke Oracle Reports by choosing the **Build Templates** menu item in the **Window** menu of the Report Configuration window. This opens the Object Navigator window which in turn provides access to the Data Model window for entering queries, and the Layout Editor for specifying the report layout, [Figure 1-7](#).

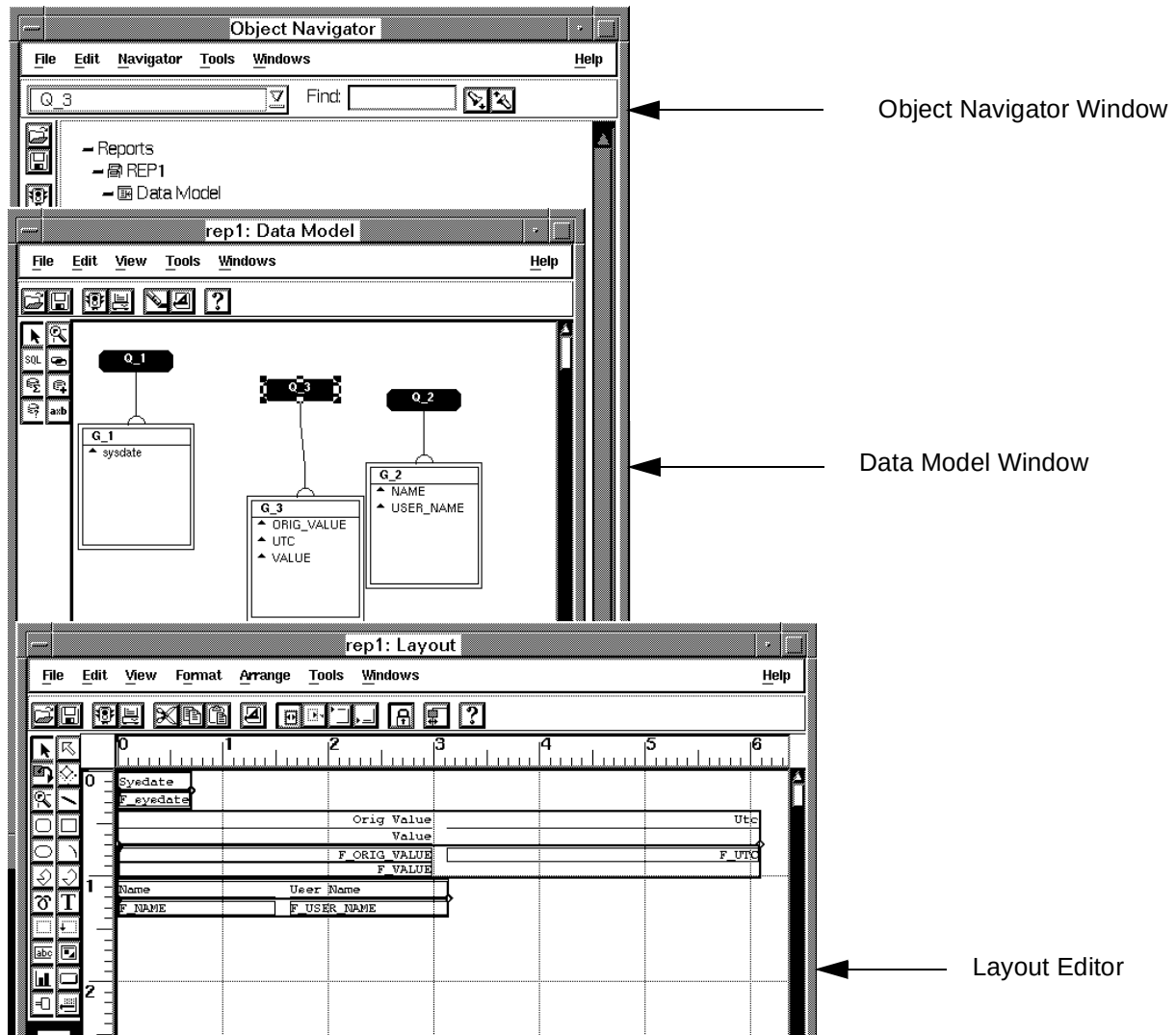


Figure 1-7. Build Template Window with Query Entered

Object Navigator

When you first invoke Oracle Reports, it creates a new report definition. The first window you see is the Object Navigator. This window displays a list of report objects. Using the Object Navigator you can easily navigate to all objects within all currently-open reports. It also provides access to the Data Model window and Layout Editor.

Data Model Window

The Data Model window is where you specify the data that you want to include in the report. The Data Model is composed of some or all of the following data definition objects:

- Queries

Queries are SQL statements that fetch data from a relational database. You can use any number of queries to select data from any number of tables (or views), located in any number of databases on any number of nodes.

- Groups

Groups determine the hierarchy of data in a report, and are used primarily to create breaks. A default group is automatically created for each query. You can create additional groups as required. For instance, if you want to create a break to group the data in the output, you can create a new group in the Data model and include the column you want to use as the break column.

- Columns

Default columns that correspond to object attributes are created automatically for you according to the SELECT statement in your query. You can create additional columns to perform computational functions on data in the default columns.

- Execution Parameters

An execution parameter is a placeholder for a variable in an SQL query. You specify the actual value for the variable when you create the report object instance. This way you can customize each execution of a report.

For example, for a Master application you may want to produce a report template that can be used to produce reports for several AI objects. Instead of using a specific object ID (that is an object name) in the template, you define an execution parameter to hold a place for the object ID. Whenever you execute the template for a specific AI object, you replace the execution parameter with the ID of the actual object.

For a MOD 300 application you may want to produce a generic batch report template that can be used to produce reports for several batches. Instead of using a specific batch identifier (that is an object name) in the template, you define an execution parameter to hold a place for the batch identifier. Whenever you execute the template for a specific batch object, you replace the execution parameter with the identifier of the actual batch.

- Data Links

Data Links are used to establish parent-child relationships between queries. For instance, a data link may be used to query for batch data which resides in more than one view.

Layout Editor

After you define the data model, you specify the report layout. You can position objects as you want them to appear in the report output. Oracle Reports provides six default layouts: tabular, master/detail, form, mailing label, and matrix. You can choose one of these default layouts and modify it as required, or create your own report layout.

1.8.3.2 Report Object Instance

The report object instance is created, by mapping execution and scheduling parameters into a report template. You can configure multiple report object instances from one template. [Figure 1-8](#) illustrates how different parameter lists and conditions can be mapped into the same report template to produce two different types of report object instances

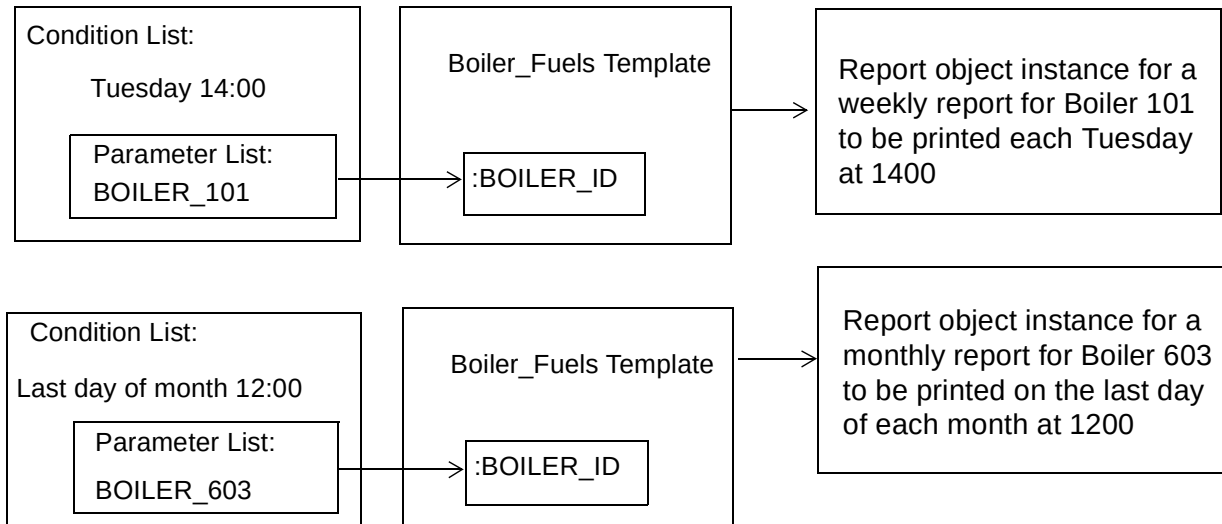


Figure 1-8. Different Report Object Instances from the Same Report Template

Creating a report object instance involves:

- assigning a name to the report object instance.
- assigning a report template to the report object instance.
- selecting scheduling modes to determine how the report can be scheduled to execute.
- configuring parameter lists that specify report template execution parameters, report output mode (Character, PostScript, PDF, HTML) and destinations (file, printer, History).
- configuring condition lists which specify the report schedule, and which parameter list to use when the report executes.

These procedures are all initiated via the Report Configuration Window.

Report Object Instance Identification

A report object instance is identified by a name, group, and description. The group is an arbitrary label that provides a means to associate related report object instances. The characters allowed are: a - z, A - Z, 0 - 9, dash (-), and underscore (_).

Report Template

By assigning a report template (built via Oracle Reports), you define the content and layout of the report object instance.

Schedule Mode

You can select one or any combination of the following schedule modes:

Manual - lets you schedule by operator command via the Manual Execution window.

Program - lets you schedule via commands from a program via AdvaInform User API, or the REPORT statement in a TCL program (TCL is only applicable for MOD 300).

System - lets you schedule by conditions specified in the Report Configuration window.

Parameter List

A parameter list specifies the following for a report object instance:

Report Output Destination This is specified via three standard execution parameters:

File - is the file where the report will be sent when executed

Printer - specifies the executed report will be printed

History - is the access name for the report log in History to store the executed report.

You can use any one or all of these parameters in combination. For instance, if you want to print the report and send the report to a report log in AdvaInform History, enable and define the Printer and History execution parameters.

Report Output Mode You can specify one of four output formats for reports:

Character Mode - Use this format when you do not require access via a web browser and the report does not contain graphics.

PostScript - Use this format when you do not require access via a web browser and the report includes graphics.

HTML or PDF - Use one of these formats when you require access via a web browser. Whether you choose HTML or PDF is a matter of preference, and depends on the software tools that you have available for web browsing. Examples of HTML and PDF report outputs are shown in [Figure 1-9](#) and [Figure 1-10](#).

Execution Parameters A report template may have parameters that serve as place holders for variables as described in [Section 3.5.2.3, How to Use Execution Parameters in a Query](#). When you create a report object instance, you must specify the values that are substituted for execution parameters when the report executes.

You must define at least one parameter list for each template and you must define at least one parameter to specify the destination for the finished report. You can define more than one parameter list for a template if you want. When you schedule the report, you specify which parameter list to use via the Condition List.

Condition List

The condition list specifies which parameter list to use when the report executes and it specifies the schedule by years, months, days, hours, minutes, and seconds. For example, you can schedule the report to execute every year, on the 1st, 15th, and last day of every month, at noon and midnight. Or, you can schedule a report to execute at 8:00 PM, every Monday, Wednesday, and Friday for January through April of the current year. For a report to execute according to the schedule defined by the condition list, the **System** scheduling mode must be enabled as described in **Schedule Mode** above.

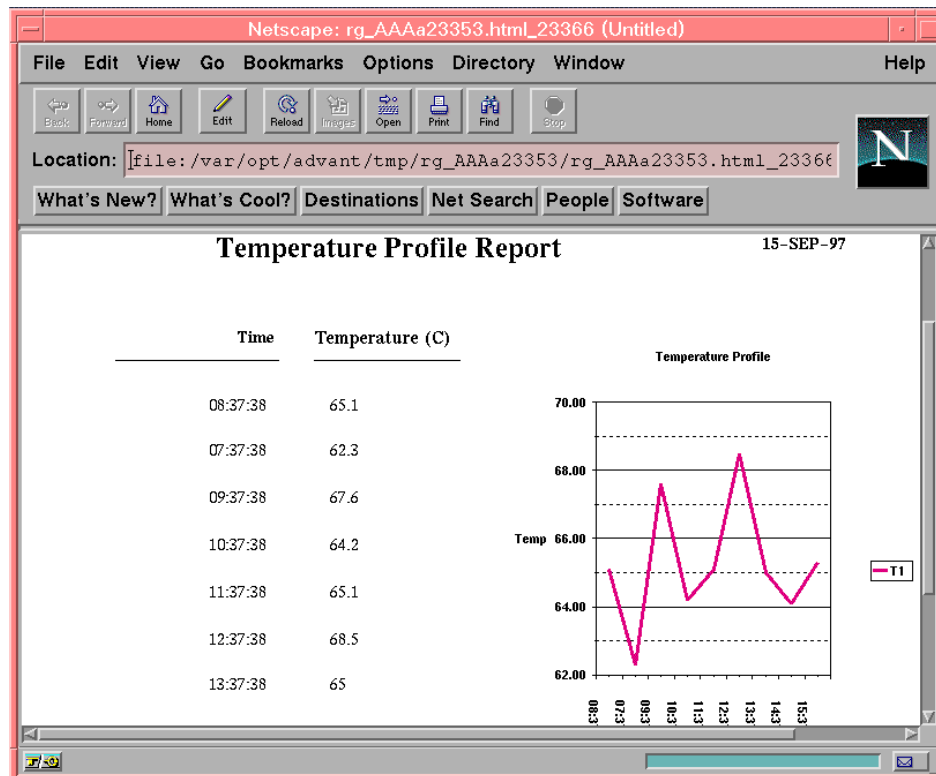


Figure 1-9. Example - Report Output in HTML Format

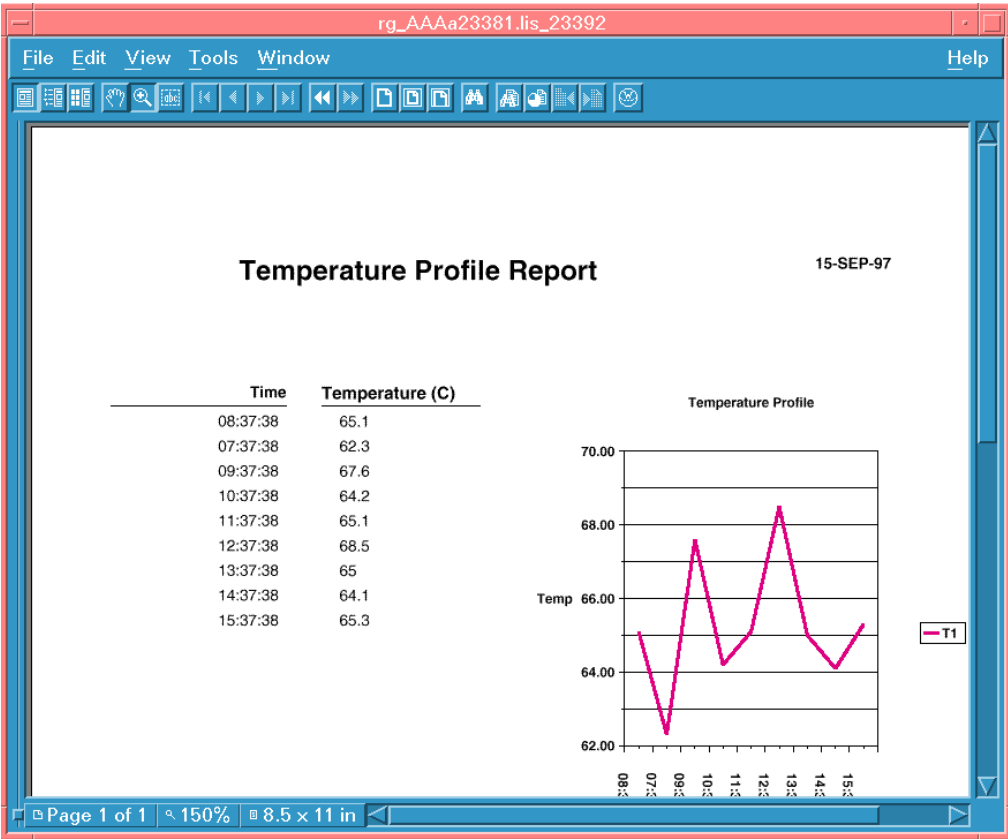


Figure 1-10. Example - Report Output in PDF Format

NOTE

Adobe® Acrobat® Reader is bundled with AdvaInform Reports software as filesset ABB-ACRO.

1.8.4 Manual Execution

You can execute a report manually via the Manual Execution window, [Figure 1-11](#).

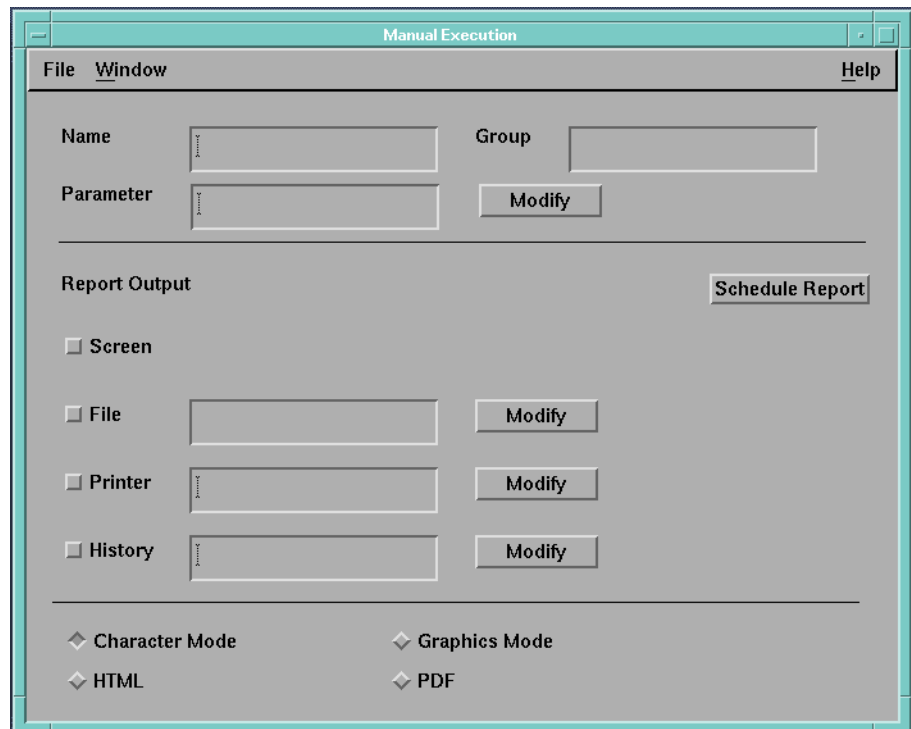


Figure 1-11. Manual Execution Window

This window operates in a manner similar to the Report Configuration window. That is you can open a specific report object instance, select a parameter list for it, and specify the destination on a one-time basis. The report executes when you click on the **Schedule Report** button.

For a report to execute according to the schedule defined by the Manual Execution window, the **Manual** scheduling mode must be enabled as described in **Schedule Mode** above. Character mode reports do not require third-party applications to be viewed on the screen or printed via the Manual Execution window. Postscript reports do not require a third-party application for printing; however, they do require a third-party application for viewing. HTML reports require a third party application for both viewing and printing. If you need the ability to launch third-party viewers and print routines, you must enter the appropriate command lines in the directory:

```
/home/opt/advant/Reports/user.environment
```

PDF reports can be viewed and printed using Adobe Acrobat Reader which is bundled with AdvaInform Reports as fileset ABB-ACRO.

1.8.5 Report Status

The Report Status window, [Figure 1-12](#), shows the status of reports in the system, and allows you to activate and deactivate reports scheduled in the program and system modes. Reports that are scheduled in the manual mode will execute even when deactivated via this window. The display is updated every ninety (90) seconds.

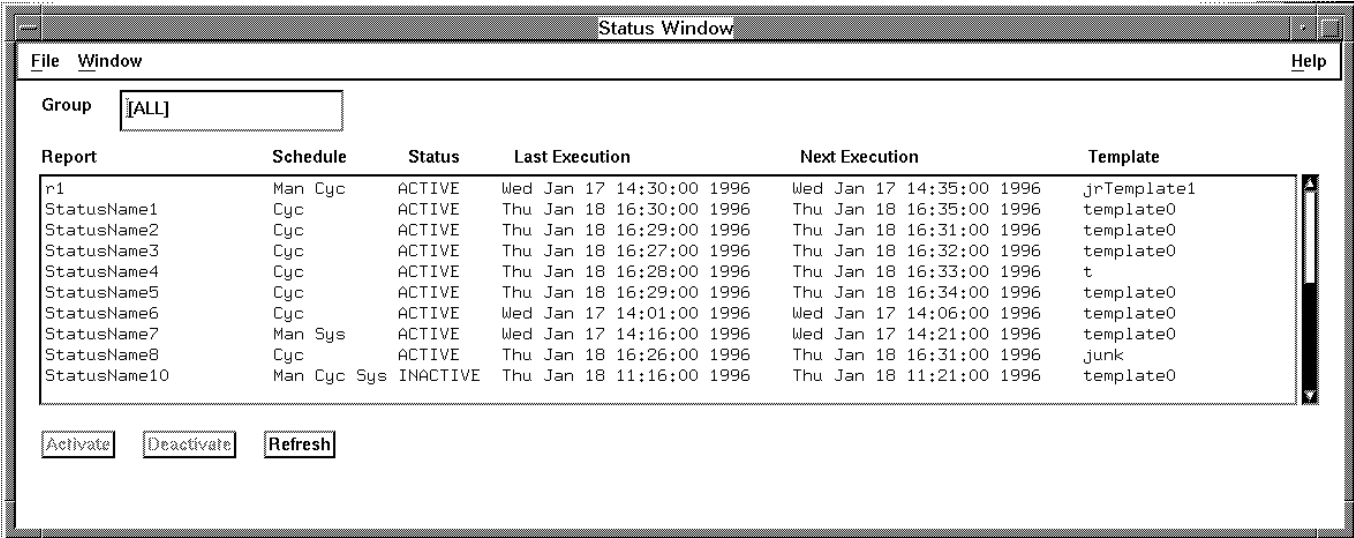


Figure 1-12. Report Status Window

1.8.6 View Report Log

AdvaInform History provides long term storage for reports. History stores reports in report logs. You can perform the following operations on report logs via the **View Logs** menu item:

- Retrieve reports from history for viewing on the screen or printing on printers.
- Modify reports and save the modified versions to history.

1.8.6.1 Viewing and Printing Reports Stored in Report Logs

Requirements for viewing and printing depend on the report output mode that was chosen when the report was executed. Character mode and PDF type reports do not require additional third-party applications to be viewed on the screen or printed. Postscript type reports do not require a third-party application for printing; however, they do require a third-party application for viewing. HTML type reports require third party applications for both viewing and printing. If you need the ability to launch third-party viewers and print routines, you must enter the appropriate command lines in the directory:

```
/home/opt/advant/Reports/user.environment
```

1.8.6.2 Modifying Character Mode Reports Stored in Report Logs

A finished report that was stored in a report log can be modified and the modified version can be saved to the report log. You can remove and edit old text, add new text, and remove and delete lines and columns. You modify report logs via a text editor such as dtpad or vi.

Data security controls govern the modifications. Any line you change is automatically designated as modified by an asterisk. The modified report is saved to the report log under a name with the suffix *.numeral*. The original report remains in its initial form. For example, if you modify SHIFT_REP, both the modified form SHIFT_REP.1 and the original form SHIFT_REP are saved on disk. If SHIFT_REP is modified a second time this second modified version is saved as SHIFT_REP.2. If SHIFT_REP.1 is modified, the new version is SHIFT_REP.1.1.

All versions of a report are kept in historical storage unless manually deleted.

1.8.7 Report Execution

A finished report results from the execution of a report object instance, [Figure 1-13](#). This process involves SQL getting the specified information from the databases. The information is formatted according to the report template. The finished report is sent to the destinations designated when the report was scheduled.

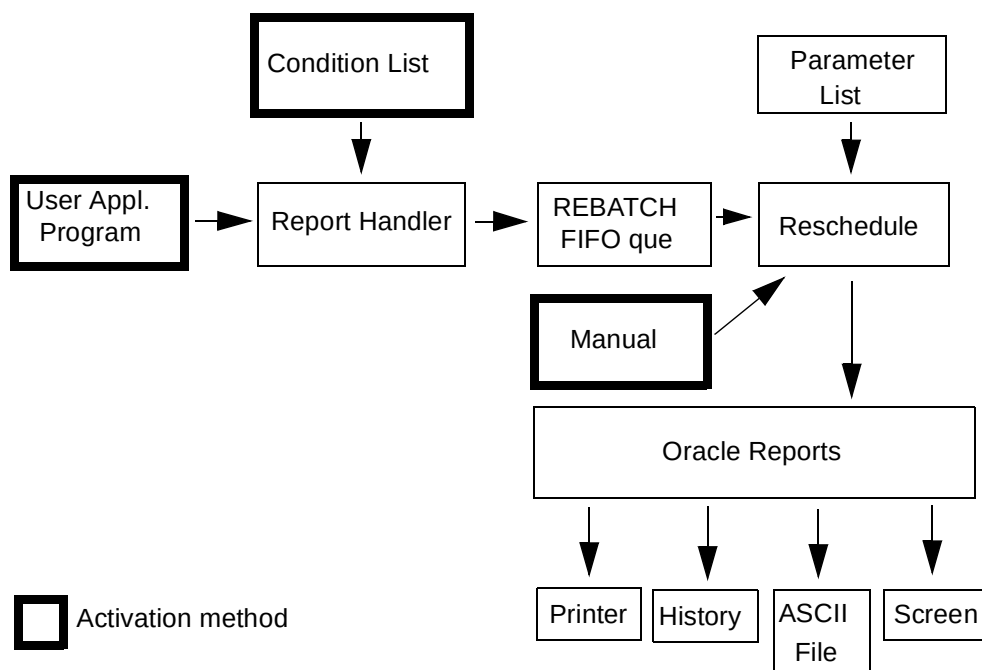


Figure 1-13. Report Execution

Reports can be printed, stored in history, or stored in a file, [Figure 1-14](#).

Requirements for viewing and printing depend on the report output mode that was chosen when the report was executed. Character mode and PDF type reports do not require third-party applications to be viewed on the screen or printed. Postscript type mode reports do not require a third-party application for printing; however, they do require a third-party application for viewing. HTML type reports require third party applications for both viewing and printing. If you need the ability to launch third-party viewers and print routines, you must enter the appropriate command lines in the directory:

```
/home/opt/advant/Reports/user.environment
```

When the report is sent to the screen or a printer, the system produces a temporary file to hold the report. The file is automatically deleted by the system. You are not notified that this temporary file exists, unless the system is short of memory and issues a message.

The report can be sent to historical storage. This provides several benefits:

- The report is stored for the long term in AdvaInform History. It can be left on the history disk or it can be archived to tape using history functionality.
- The report can be retrieved from the log for viewing on the screen or for printing.
- The report can be modified with the modifications being controlled by a data security feature as described in [Section 1.8.6.2, Modifying Character Mode Reports Stored in Report Logs](#).

If you are using the encapsulation feature which allows you to insert data from a finished report into another report, the data you specify for encapsulation is saved to the same report log as the finished report.

The report can be sent to a file in a text, HTML, or PDF format. The file is outside the Reports domain. The report cannot be brought back into the Reports package from the file. There is no control over any modifications that are made to the file.

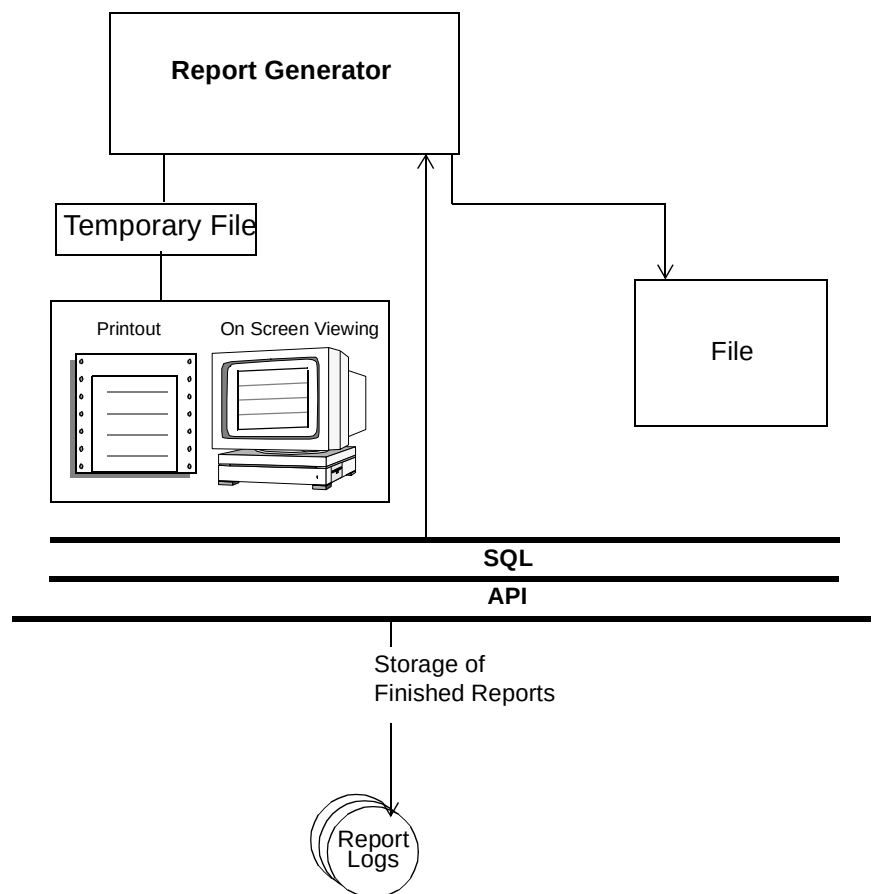


Figure 1-14. Report Destinations

Chapter 2 Installation

2.1 Site Planning Environment

This section is not applicable.

2.1.1 Guidelines for planning installation of Advant Station

Refer to the *Advant Station 5xx Hardware User's Guide*.

2.2 Setup

Generally, AdvaInform Reports software is loaded onto the Advant Station at the factory prior to delivery. Therefore, you are generally not responsible for loading the software. If you are required to load the software for some reason (for instance, if you purchase AdvaInform Reports software after initial delivery of the Advant Station), refer to the *Advant Station 500 Series IMS User's Guide* (or *Advant Station Operator Station User's Guide*) for instructions.

For hardware, refer to *Advant Station 5xx Hardware User's Guide*.

2.3 Start-up Procedures

The AdvaInform Reports run time environment is automatically started as part of the Advant Station start-up. This procedure is covered in the *Advant Station Information Management Station User's Guide* and *Advant Station Operator Station User's Guide*.

When you use AdvaInform Reports on an Advant Station with AdvaInform software, the Report Configuration window is activated from the IMS menu. How to open this window is described in [Section 3.3, Application Start-up](#).

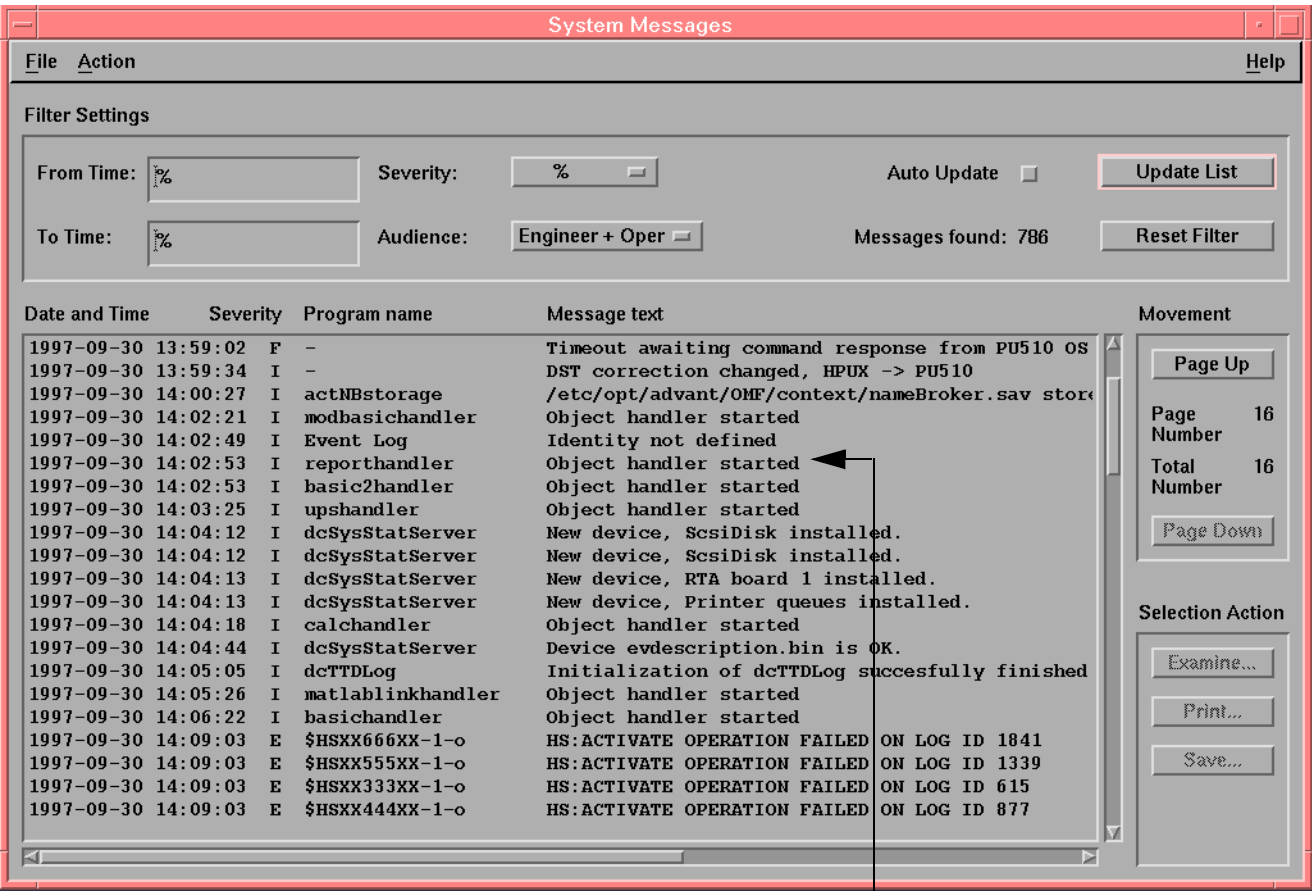
When you use AdvaInform Reports on an Advant Station with AdvaCommand software, External Display Access must be configured in the Environment Builder to support Reports windows. Refer to the *Environment Builder User's Guide* for details.

2.4 Product Verification

From the Report Configuration Window, choose **Help - On Version** from the **Help** menu. Verify that the version corresponds with the delivery information. Choose **Exit** from the **File** menu to close the Report Configuration window.

Next, open the Report Status window as described in [Section 3.3, Application Start-up](#). No reports should be executing.

To verify that the software is properly installed and running, check the System Messages display as described in the *AdvaInform Basic Functions User's Guide*. The process “reporthandler” should be up and running, [Figure 2-1](#).



line for reporthandler process

Figure 2-1. System messages Display

Chapter 3 Configuration and Application Building

3.1 Design Considerations

Few limitations exist in setting your parameters and times in AdvaInform Reports. This lets you customize your reports for your situation. The wide range of options makes it possible for you to over or under configure the amount of data that you include in your reports. Therefore, it is recommended that you re-check your configurations if performance is not satisfactory.

3.1.1 Printer Considerations

During installation, a default printer with the name ReportPrinter is automatically configured. This is required for graphic mode execution. It is recommended that you re-configure the ReportPrinter using the HP-UX SAM utility to match the printer that will be used to print reports. You can change any of the configuration parameters, except the name which **MUST** be: ReportPrinter.

3.1.2 AdvaInform History Considerations

AdvaInform History supports AdvaInform Reports in the following ways. It is a source of historical data for reports. In addition, it provides storage for finished reports, allows finished character-based reports to be modified while maintaining data integrity, and allows information from one report to be inserted into another report.

3.1.3 User API Considerations

Since the Report instances are seen as objects they can be accessed as any object through AdvaInform User API. The interfaces to be used are the Execute and Status Statements which are described in the *AdvaInform User API User's Guide*. Also refer to the *AdvaInform Object Types Reference Manual* for a detailed description of the Report object attributes.

3.1.4 Batch/TCL Considerations (MOD 300 Only)

The following Taylor Control Language (TCL) statements support Batch reports: Report, Start_Batch, End_Batch, and Record. These statements are described in the *TCL User's Guide*.

The following requirements must be met to enable activation of a report from a TCL program:

- When scheduling parameters are configured in the Report Configuration Window, the **Program** option must be selected. See [Section 3.6.2, Report Configuration Window](#).
- A parameter list called TCLlist must be defined. See [Section 3.6.2.1, Parameter List Configuration Dialog](#).
 - The parameter list must use the name `TCLlist` exactly as shown.
 - If you choose to save the report in a report log (in AdvaInform History), you must enter the name of the report log in the History field of the Parameter List Configuration dialog box.
 - Two parameters, *batchid* and *unitid*, must be defined and initialized at meaningful values (parameter names must be defined as batchid and unitid). These values serve as place holders. The actual runtime values for these parameters are provided by TCL. The values are either text strings defined in the REPORT statement (see *TCL User's Guide*) or, if the parameters are undefined in the REPORT statement, they are names of the batch and unit that ran the TCL sequence.

3.1.5 Building the AdvaInform SQL*Connect Location Table

AdvaInform SQL*Connect allows you to use SQL queries to retrieve data from Advant OCS objects. To do so, the AdvaInform SQL*Connect Location Table must contain the names of the objects to be queried. This procedure is described in the *AdvaInform SQL*Connect User's Guide*.

3.1.6 Building Report Logs

To store reports via AdvaInform History, you must build report logs for the reports. This procedure is described in the *AdvaInform History User's Guide*. When you build the report logs, remember the following:

- Report logs are primary logs.
- The collection mode for a report log must be Asynchronous.
- The Log Access Name is used to identify a report log when you send a report to it.

3.1.7 Report Output Mode

When you execute a report, either automatically according to a schedule, or manually, you specify the report to be output in any one of the following formats:

Character	Use this format when you do not require access via a web browser and the report does not contain graphic images (bitmaps). Character mode reports do not require third-party applications to be viewed on the screen or printed.
PostScript	Use this format when you do not require access via a web browser and the report includes graphic images. Graphic mode reports do not require a third-party application for printing; however, they do require a third-party application for viewing.
HTML or PDF	Use one of these formats when you require access via a web browser. HTML and PDF reports require third party applications for both viewing and printing. Whether you choose HTML or PDF is a matter of preference, and depends on the software tools that you have available for web browsing.

NOTE

Printer is not supported as a destination when the output mode is HTML.

You should consider which report output mode you will use as you design the report layout. The different report output modes have different capabilities and limitations.

When you build a report template using Oracle Reports, you must choose to build the template in either the Character mode or Graphics mode. Base your choice on the following:

- Whether you build the template in the Character mode or Graphics mode DOES NOT affect your ability to execute and store reports in HTML or PDF format. Reports whose report templates were created in either Character or Graphics mode can be stored in HTML or PDF format.
- If there is any possibility that you will use graphic images in the report in the future, you must start building the report template in Graphics mode. You can change a report template that you began in Graphics mode to Character mode if you find that you will not use graphic images. However, once you choose Character mode, you can not change to Graphics mode.

The default mode for Oracle reports is Graphics mode. Since Oracle Reports defaults to Graphics mode, if that is the mode you want to use, there are no special procedures with regard to mode selection. On the other hand, if you want to use Character mode, you must set this mode as described in [Section 3.3.3, How to Configure the Report Output for Character Mode](#).

The choice to execute a report in either of the four modes is made when you schedule the report. The only limitation is that reports whose templates were created in the Character mode can not be executed in the Graphic mode, and reports whose templates were created in the Graphic mode can not be executed in the Character mode.

3.1.8 Making Reports Accessible on the Web

Reports that are output in HTML or PDF format can be made available to PC clients equipped with web browsers by storing the reports in a directory that is accessible from a web server.

The Netscape FastTrack web server can be installed on an Advant Station with AdvaInform software at no charge. Installation and set up instructions are provided in the *Advant Station 500 Series IMS User's Guide* and *AdvaInform Basic Functions User's Guide*.

3.2 Capacity and Performance

Capacity and performance data for this release is provided in the *Advant Station 500 Series IMS User's Guide*.

3.3 Application Start-up

You can access AdvaInform Reports from the Advant Station configured to support Reports. After you gain access to Reports windows, the operation of all Reports functions is the same whether you are using an Advant Station with AdvaInform software or AdvaCommand software.

3.3.1 Starting Reports on Advant Station w/AdvaCommand (MOD 300 Only)

To run Reports on an Advant Station with AdvaCommand software, External Display Access must be configured in the station environment to support Reports windows. This procedure is described in the *Release Notes for the Advant Station 500 Series Operator Station with AdvaInform and TCL Builder Options*. It is also described in the *AdvaBuild Environment Builder User's Guide*.

The script for configuring external display access for AdvaInform Reports is:

```
$RG_HOME/etc/exerereport $@
```

Instructions for navigating windows in the Advant Station with AdvaCommand software are provided in the *AdvaCommand Basic Functions User's Guide*.

3.3.2 Starting Reports on Advant Station w/AdvaInform

This section describes how to open the Reports Configuration window from the IMS menu. For instructions on how to log on to the Advant Station and open the IMS menu, refer to the *Advant Station 500 Series IMS User's Guide*.

Access to AdvaInform Reports is via the **AdvaInform** menu, [Figure 3-1](#).

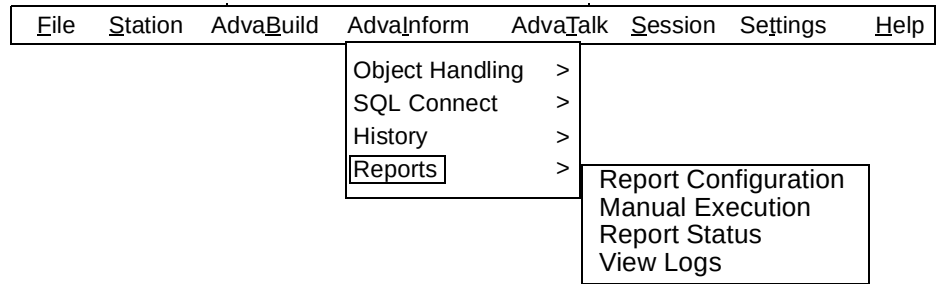


Figure 3-1. AdvaInform Menu Items

To access any of the reports display windows from this initial display:

1. Choose **Reports** from the **AdvaInform** menu. This displays a submenu with Report options.
2. Choose one of these options.

3.3.3 How to Configure the Report Output for Character Mode

Since Oracle Reports defaults to Graphics mode, if that is the mode you want to use, you do not have to do anything. On the other hand, if you want to use Character mode, you must set this mode as described below.

CAUTION

If you plan to use Character mode, be sure to change the mode from Graphics to character BEFORE you begin building the report.

To change the report output mode from graphic to character:

1. Open the Object Navigator, [Figure 3-2](#), by choosing **Build Templates** from the **Window** menu in the Report Configuration window.

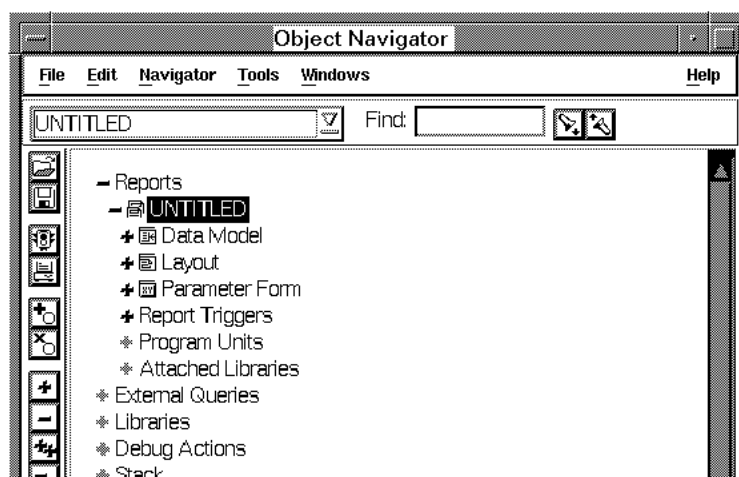


Figure 3-2. Build Template Window Shown w/Query Entered

2. Double-click on the icon next to the report name, [Figure 3-3](#).

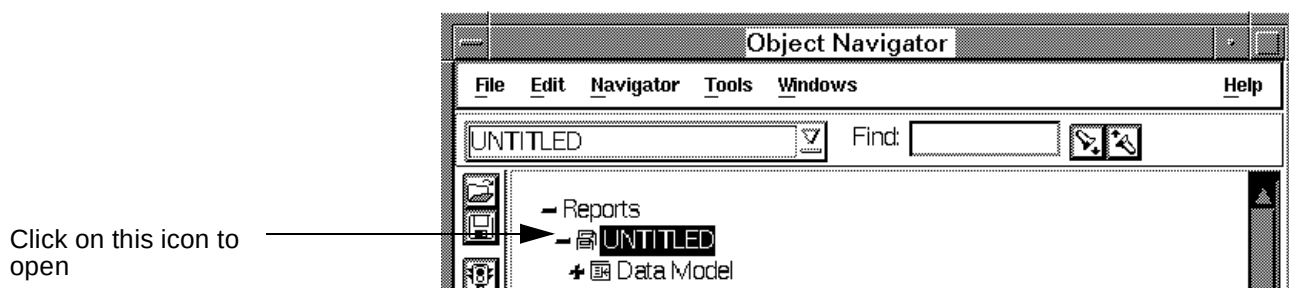


Figure 3-3. Report Icon

This displays the Report Properties dialog box for the selected report, [Figure 3-4](#).

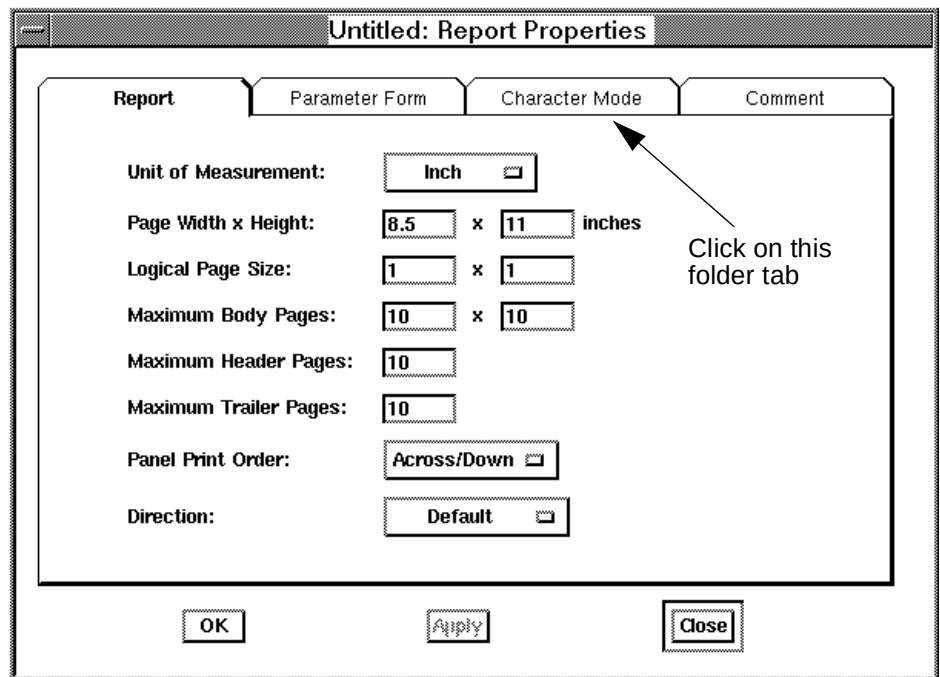


Figure 3-4. Report Properties Dialog Box

3. Click on the **Character Mode** folder tab.

This changes the list of properties displayed in the dialog box, [Figure 3-5](#).

The screenshot shows a dialog box titled "Untitled: Report Properties". It has four tabs: "Report", "Parameter Form", "Character Mode" (which is selected), and "Comment". The "Character Mode" tab contains the following settings:

- Report Width x Height: 80 x 66
- ☐ Previewer Hint Line: [text box]
- ☐ Previewer Status Line: [text box]
- ☐ Use Character Units in Designer ← Select this button
- ☐ Convert Bitmapped Objects to Boxes
- ☐ Convert Borders
- ☐ Disable Host
- ☐ Disable Split
- ☐ Disable Zoom
- ☐ Start in Zoom
- ☒ Suppress Previewer Title

At the bottom of the dialog box are three buttons: "OK", "Apply", and "Close".

Figure 3-5. Character Mode Properties

4. Select the button labeled: **Use Character Units in Designer**.
5. Click **OK** and then click **Close**.

3.4 Tutorial

This tutorial takes you step-by-step through the procedures to produce a report for a numeric log from AdvaInform History. The query for this report will return the log name, value, and time stamp attributes which are stored in the NUMLOGVAL view in the Oracle database. A time range will be specified to limit the data returned by the query. Execution parameters will be inserted in the query to serve as place holders for the start and end times of the time range. The actual start and end times will be entered as parameters in the parameter list when you create the Report Object instance. When the report is executed it will produce a report on the screen as shown in [Figure 3-6](#).

Value	Name	TimeStamp
0	\$HSAIC72_2, VALUE-1-o	09/05/96 14:06:09
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:06:19
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:06:29
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:06:38
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:06:47
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:06:56
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:07:05
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:07:14
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:07:23
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:07:32
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:07:41
20	\$HSAIC72_2, VALUE-1-o	09/05/96 14:07:50

Figure 3-6. Example Numeric Log Report

This tutorial covers the following basic procedures:

- How to open the Report Configuration window. This window provides access to all the tools required to build a report.
- How to build a report template using Oracle Reports. This tutorial provides a quick hands-on introduction to Oracle Reports. For further information regarding Oracle Reports procedures, particularly report formatting, refer to the applicable Oracle Reports documentation as described in [Section 1.5, Related Documentation](#).
- How to create a report object instance which involves mapping execution and scheduling parameters to the report template

NOTES

During this tutorial, you may find that the contents of some windows and dialog boxes in your system may differ from the windows and dialog boxes shown in this book. This is because you may already have report templates and report object instances defined in your system. Also, you may want to use report and parameter names that are more meaningful for your particular application than the ones suggested in this tutorial.

3.4.1 Open the Report Configuration Window

To open the Report Configuration window:

- 1. Choose **Reports** from the **AdvaInform** menu as described in [Section 3.3, Application Start-up](#). This displays a submenu with Report options, [Figure 3-7](#).

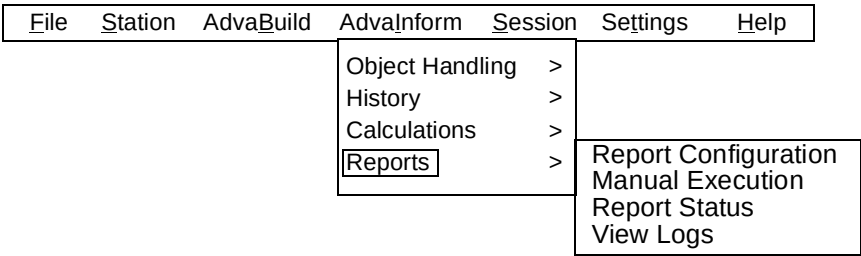


Figure 3-7. Reports Submenu

- 2. Choose the **Report Configuration** option from the submenu. This opens the Report Configuration window, [Figure 3-8](#).

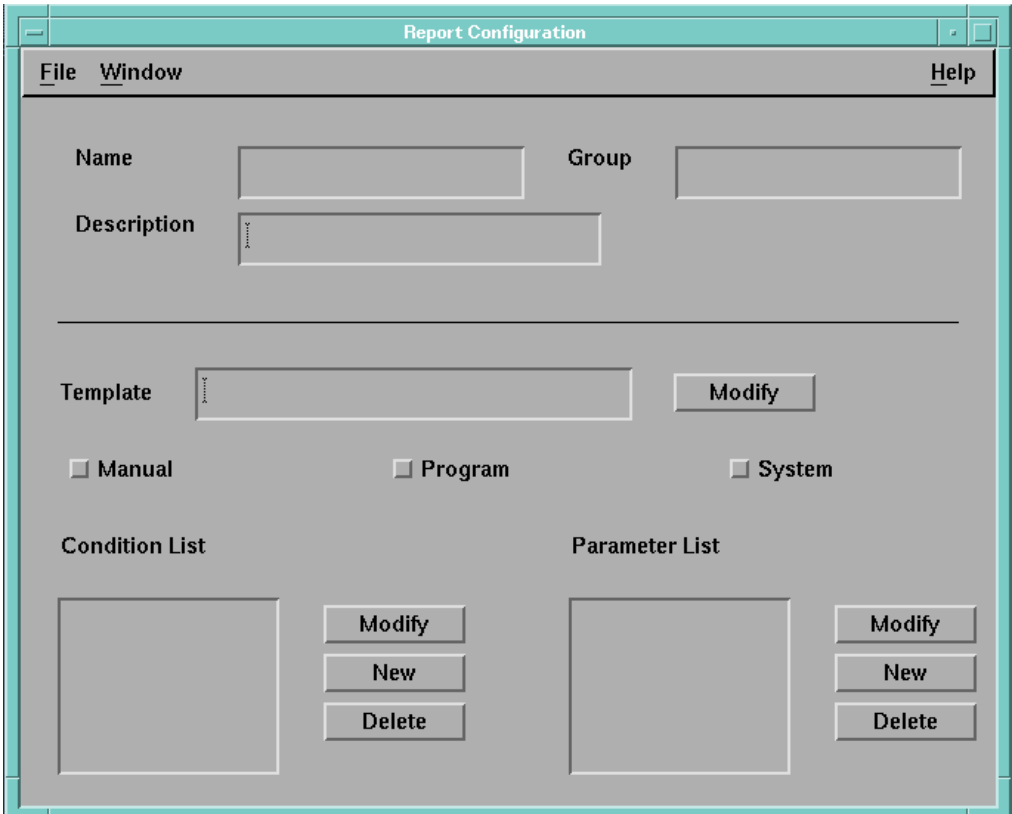


Figure 3-8. Report Configuration Window

Initially, the window is empty.

3.4.2 Build the Report Template

The report template defines the contents and format of the report. You build the report template via Oracle Reports which is embedded in AdvaInform Reports. This tutorial demonstrates some basic operations in Oracle Reports. For further information regarding Oracle Reports, (particularly, how to format a report template) refer to the applicable documentation as described in [Section 1.5, Related Documentation](#).

3.4.2.1 Open the Object Navigator

To open the Object Navigator, [Figure 3-9](#), choose **Build Templates** from the **Window** menu.

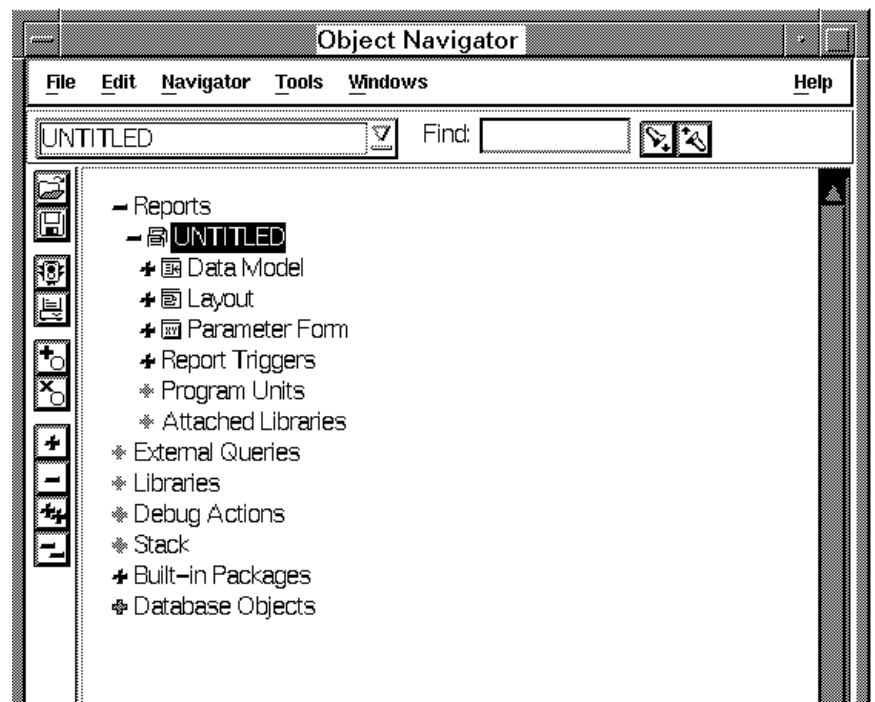


Figure 3-9. Oracle Reports Object Navigator

When you open this window initially, it creates the objects for a new report named UNTITLED. These objects include a Data Model, Layout, Parameter Form, Report Triggers, Program Units, and Attached Libraries. From here you can either begin building a new report, or open an existing report.

3.4.2.2 Select the Output Mode for the Report

Since this tutorial does not include graphic images and requires you to output your report to the screen, the report template must use Character mode. To select Character mode, refer to [Section 3.3.3, How to Configure the Report Output for Character Mode](#).

3.4.2.3 Create a New Report Template

To create a new report template:

1. Double-click on the Data Model Icon, [Figure 3-10](#).

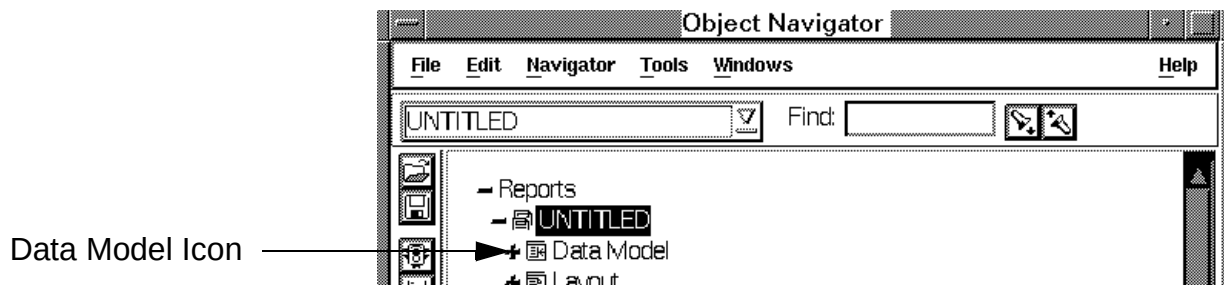


Figure 3-10. Data Model Icon

This opens the Data Model window for your new (Untitled) report, [Figure 3-11](#).

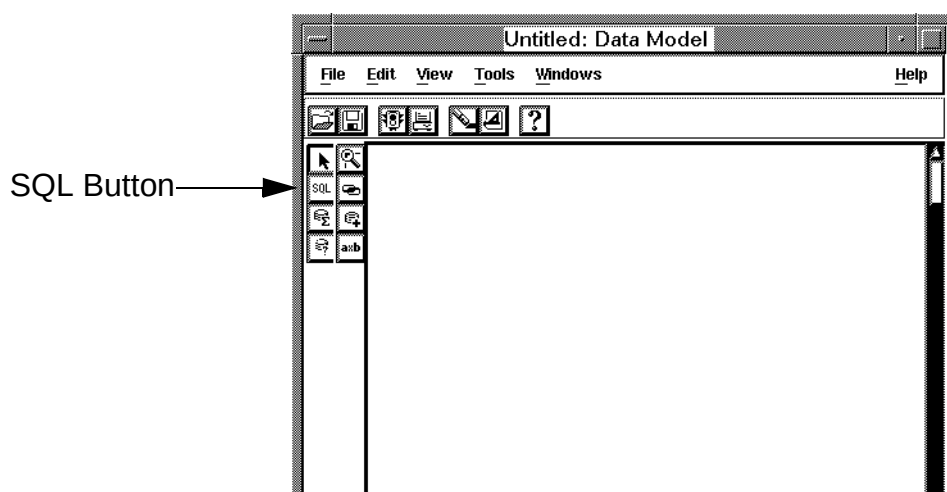


Figure 3-11. Data Model Window

The Data Model window is where you specify the data you want to retrieve with your query.

NOTE

When you move the cursor over a button on this window and leave the cursor there, help text that describes the function of the button is displayed.

2. Click once on the SQL button, [Figure 3-11](#), and then drag the cursor into the work area in the Data Model window.

The cursor becomes a cross hair.

3. Position the cursor where you want to place the query and then click.

This creates a query object represented by a rounded rectangle in the Data Model window, and opens the Query Properties Sheet, [Figure 3-12](#).

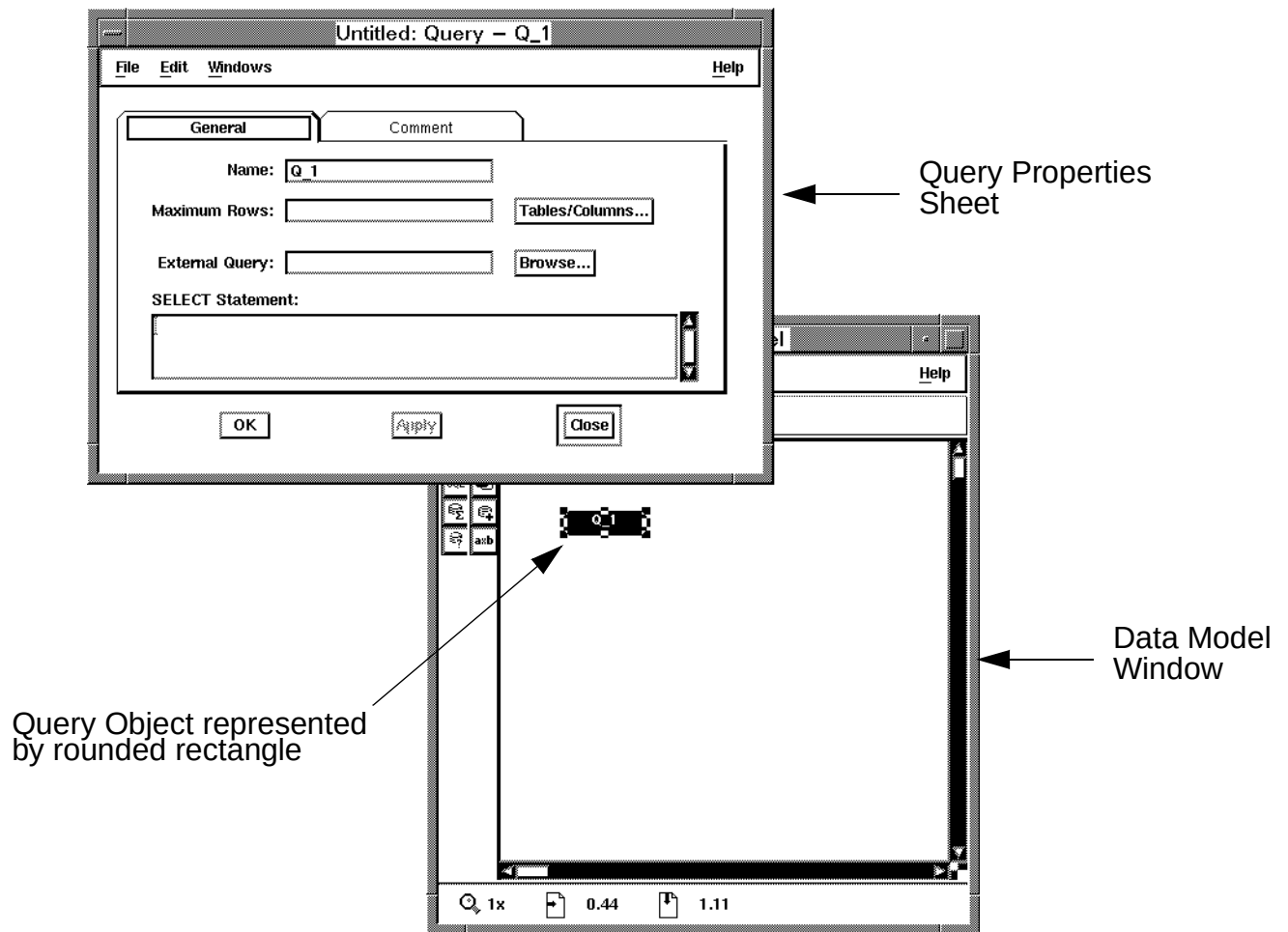


Figure 3-12. Query Object and Query Properties Sheet

The query is assigned a default name Q_n where n is the sequential number for the query. For instance, the first query is assigned the name Q_1. The second query will be assigned the name Q_2, and so on.

You can assign a new name using the Name field on the Property Sheet if you want; however, this is not mandatory. The only requirement is that each query have a unique name.

4. To begin defining the query, click on the **Tables/Columns** button on the Properties Sheet.

This displays the Table and Column Names dialog box, [Figure 3-13](#).

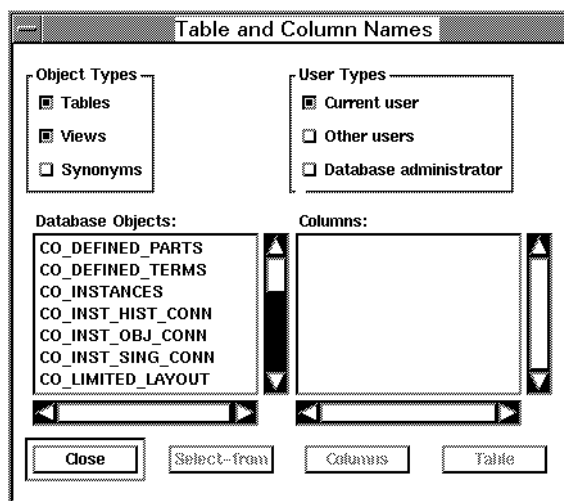


Figure 3-13. Table and Column Names Dialog Box

This dialog box makes it easy for you to find the pre-defined views in the Oracle database, and lets you select both the view (object type) and the column (object attribute) that you require. The database object you select in this dialog determines the view to be used in the FROM clause of your query. The Column you select in this dialog determines the object attribute(s) to be used in the SELECT clause of your query.

NOTE

If you prefer, you can skip this dialog box, and enter the entire query yourself directly in the SELECT Statement part of the Query Property Sheet.

5. To continue using the Tables and Columns dialog box, deselect **Tables** in the Object Types part so that just Views is selected. Also, select **Other users** in the User Types part so that both Current and Other Users are selected, [Figure 3-14](#).

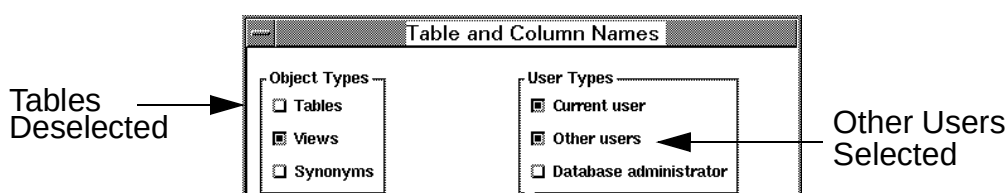


Figure 3-14. Tables and Column Names Selection

Deselecting Tables limits the scope of the database objects to just views. Selecting Other users, includes the NUMLOGVAL view that you require for this tutorial. In this case, users does not refer to AdvaInform users, but rather an Oracle class of users.

NOTE

When you use the Table and Column Names dialog to define your own queries, do not use tables or views with the user prefix "PUBLIC". Instead, find the matching table or view with the user prefix "GTADMIN".

6. Scroll down the list of database objects to GTADMIN.NUMLOGVAL and select it.

The GTADMIN prefix is the Oracle user that owns the NUMLOGVAL view. If you enter the query directly in the Query property Sheet, you are not required to use this prefix.

When you select the NUMLOGVAL view, the columns available in this view are presented in the Columns selection window, [Figure 3-15](#).

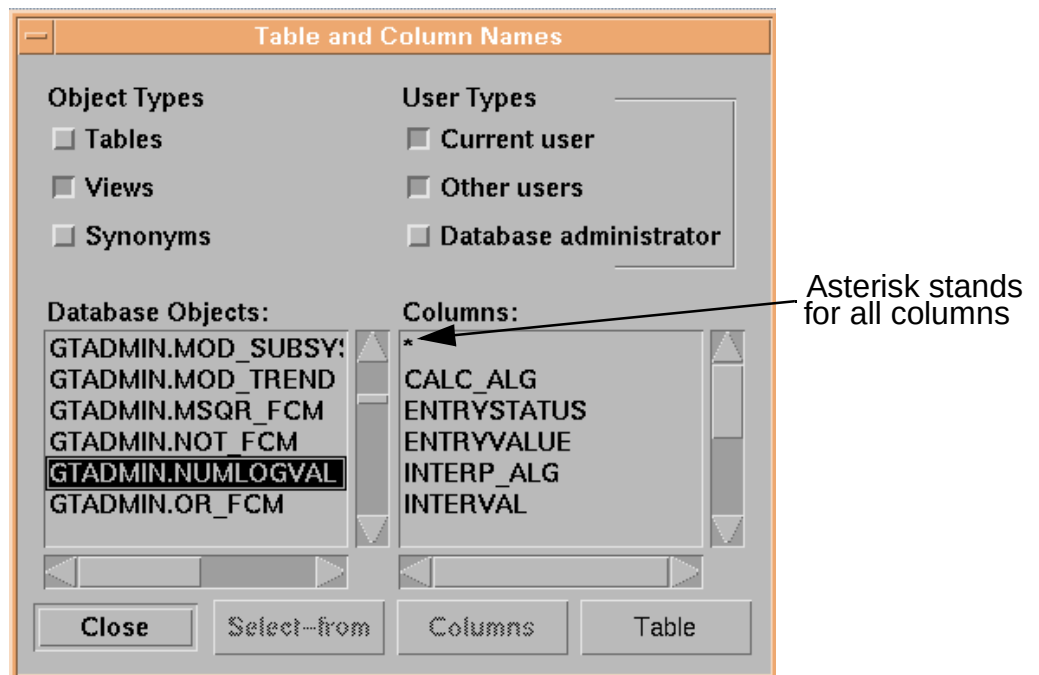


Figure 3-15. Columns in NUMLOGVAL View

The asterisk at the top of the list stands for all columns. Selecting the asterisk would put all column names for NUMLOGVAL in the SELECT clause of your query.

For this tutorial, select ENTRYVALUE, NAME, and TIME. This will provide the name of the numeric log, the entry value, and the corresponding time stamp.

When you select a column name, the **Select-From** button becomes active.

NOTE

If you select a column by mistake, you can deselect it by clicking on it again.

7. When you have selected the columns you need, click on the **Select-From** button.

This enters the SELECT and FROM clauses for your query in the Query properties sheet, [Figure 3-16](#).

SELECT Statement:

```
select ENTRYVALUE, NAME, TIME from GTADMIN.NUMLOGVAL
```

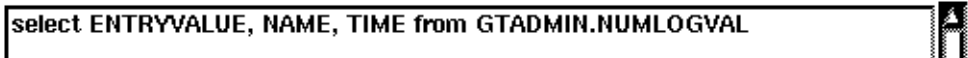


Figure 3-16. Updated Query Properties Sheet

The Table and Column Names dialog box remains open if you want make additional selections.

8. Click on **Close** to remove this dialog box.
9. Continue entering your query directly in the Query Property Sheet now by specifying a name and time range to limit the data returned by your query.

NOTE

The TIME attribute is a date data type. When you use TIME alone it provides the day, month, and year. To get the full time stamp, use the to_char function with TIME to convert TIME from a date data type to a character data type.

Insert the to_char function as shown in [Figure 3-17](#). When you convert time to a character data type, you must specify the format mask. In this case the mask is 'mm/dd/yy hh24:mi:ss'. This stands for months, days, year, and hours, minutes, and seconds in 24-hour clock format. Other formats are possible. For further information see [Section 3.7.6.2, Basic Query for Numeric Data](#).

SELECT Statement:

```
select ENTRYVALUE, NAME, to_char(TIME, 'mm/dd/yy hh24:mi:ss')  
from GTADMIN.NUMLOGVAL
```

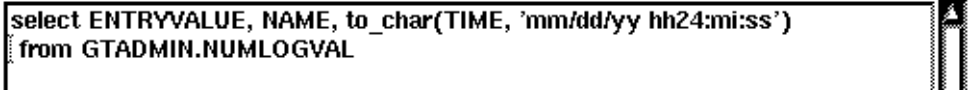


Figure 3-17. Inserting the to_char function

Enter the WHERE clause as shown in [Figure 3-18](#).

NOTE

Rather than use the log name shown in the example below, substitute a log name that fits your system.

SELECT Statement:

```
select ENTRYVALUE, NAME, to_char(TIME, 'mm/dd/yy hh24:mi:ss')  
from NUMLOGVAL  
where NAME = '$HSAIC72_2,VALUE-1-o'  
and time>=to_date(:STIME, 'mm/dd/yy hh24:mi:ss')  
and time<=to_date(:FTIME, 'mm/dd/yy hh24:mi:ss')
```

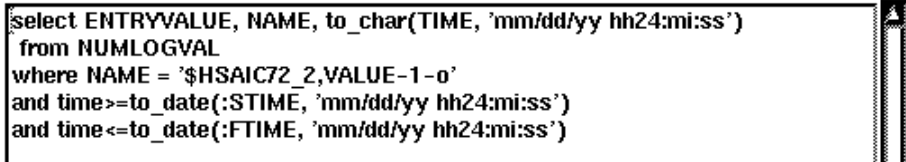


Figure 3-18. Completed Query

STIME and FTIME are execution parameters that represent the start and end of the time range for your query. Execution parameters are place holders for actual values. The values for these execution parameters must be specified in report object instance before you run the report. The colon (:) before the text strings STIME and FTIME mark them as execution parameters.

The STIME and FTIME values are passed into the query as *character* data types, and must be converted to *date* data type in order to be compared to *time*. This conversion is performed by the `to_date` function.

10. When you are finished with the query, click **OK**.

When your query includes an execution parameter, you get a message similar to the one shown in [Figure 3-19](#). Since this query has two execution parameters, you will get two such messages, one-message-at-a-time.

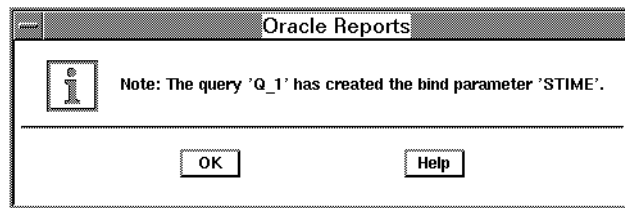


Figure 3-19. Parameter Message

11. Click **OK** to acknowledge the first message. When the next message is displayed, click **OK** again.

This closes the Query Property Sheet and updates the query object in the Data Model window to show the object attributes included in the query, [Figure 3-20](#).

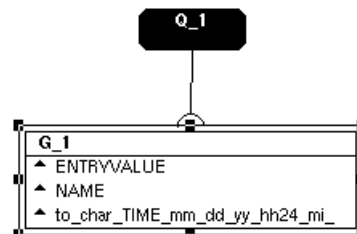


Figure 3-20. Updated Query Object

This is a good time to save your work. This way, if you inadvertently exit the application, your work up to this point will be saved.

12. To save the report template, choose **Save** from the **File** menu in the Data Model window.

This displays the Save dialog, [Figure 3-21](#).

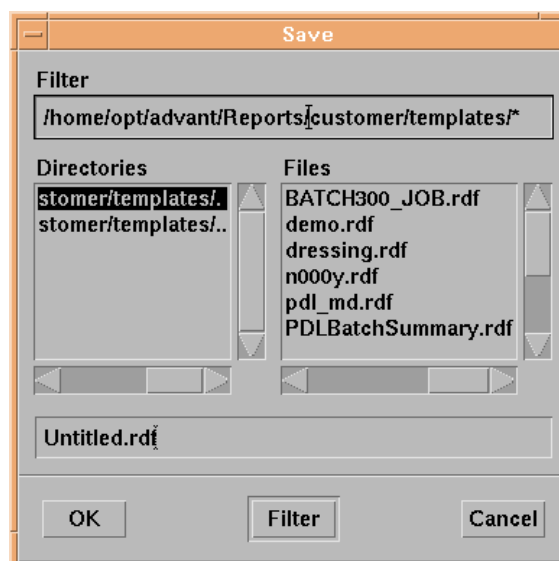


Figure 3-21. Save Dialog

CAUTION

The default directory specified in the Filter field is where the report execution function looks for report templates for scheduling and so on. If you change the path and store them in another directory, the report execution function will not find them. Therefore, DO NOT change the default directory specified in the Filter field.

13. Enter a unique name for the template in the field near the bottom of the dialog box. Give the name an **rdf** extension. For example:

tutorial.rdf

14. Click on **OK**.

This closes the save dialog and saves the new template.

Now that you have specified what data you want to include in your report, the next step is to specify how you want the data to look. This is done via the Reports Layout window. You can either start with a default layout or define a new layout from scratch. This tutorial uses the default tabular layout.

15. To open the Reports Layout window choose **Default Layout** from the **Tools** menu in the Data Model window.

This displays the Default Layout dialog box, [Figure 3-22](#).

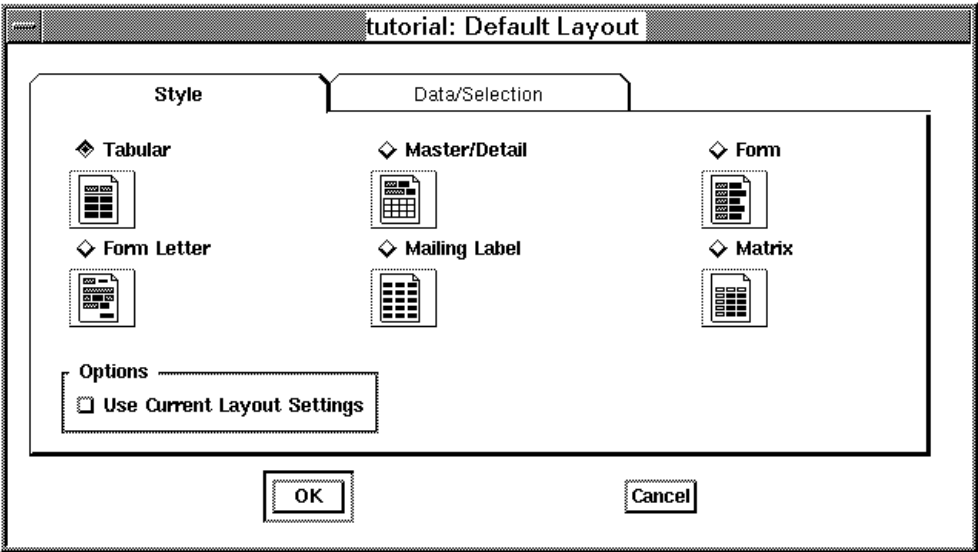


Figure 3-22. Default Layout Dialog Box

The tabular format is the default selection.

- 16. Select the Tabular format, if it is not already selected.
- 17. Click on the **Data/Selection** Tab.

This displays the Data/Selection page of the Default Layout dialog box, [Figure 3-23](#).

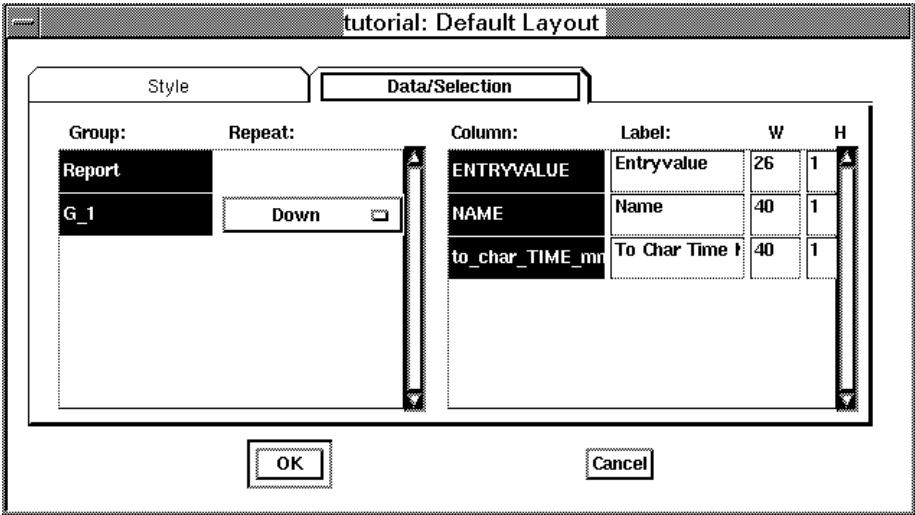


Figure 3-23. Data/Selection Page

This page provides the means for you to re-name and size your report columns. The default column labels and sizes may not be meaningful or practical. For instance the column label for the TIME attribute is to_char_TIME..., and the width of all the columns is too large to fit on a single line.

18. Re-name and re-size your columns as described below and shown in [Figure 3-24](#).
- For ENTRYVALUE, re-define the label as Value, W = 12
 - For NAME, re-define the label as Name, W = 20
 - For to_char_TIME..., re-define the label as TimeStamp, W = 21.

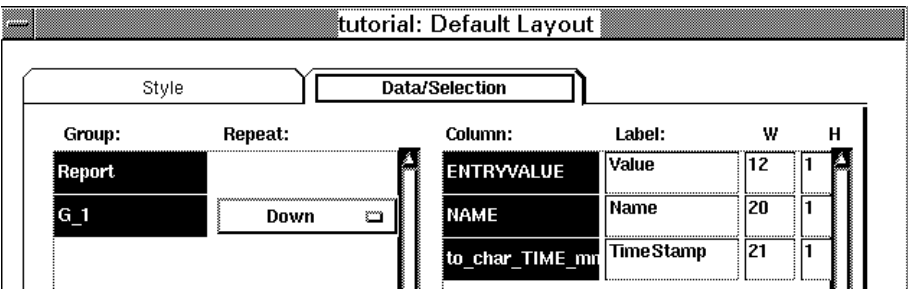


Figure 3-24. Re-naming and Re-sizing Columns

19. Click **OK** when you are finished with this dialog.
- This opens the Reports Layout window in the tabular format, [Figure 3-25](#).

Run Report Icon

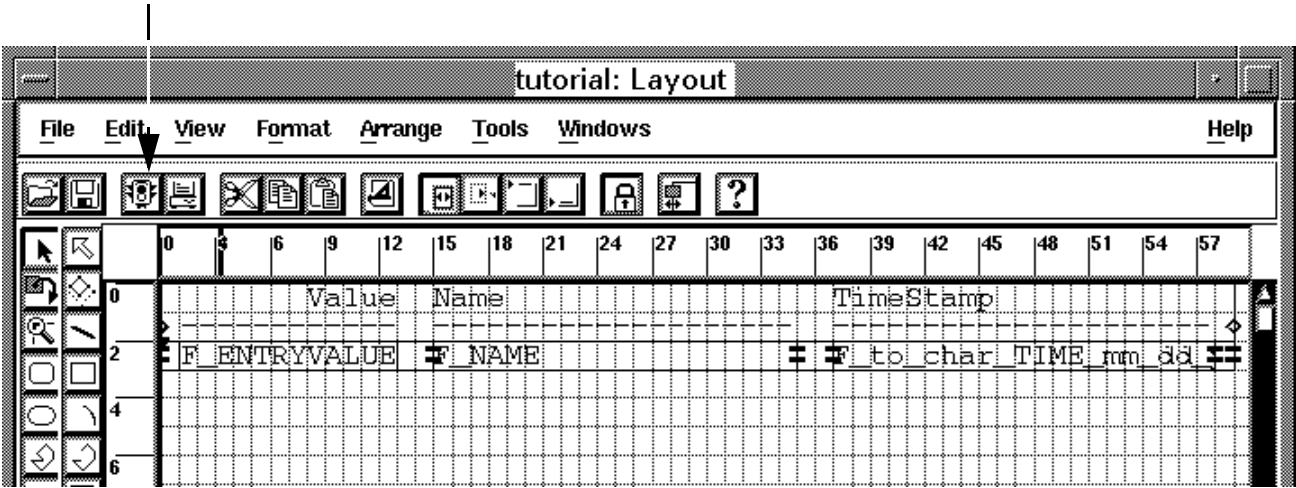


Figure 3-25. Report Layout Window

This default layout is adequate for this tutorial. It provides column labels based on the names of the object attributes that you specified in your SELECT statement.

20. Save the report template by choosing **Save** from the **File** menu in the Layout window.

NOTE

You must save the report template from the Layout window at least once; otherwise, you will get an empty file when the report is executed. After you save the report once from the Layout window, you can save it either from the Layout window or Data Model window for subsequent saves.

Now you can preview your report to verify that it retrieves and presents the data in the manner that you intended.

21. To preview the report, click on the Run Report icon which looks like a traffic light, [Figure 3-25](#).

This opens the Runtime Parameter Form, [Figure 3-26](#).

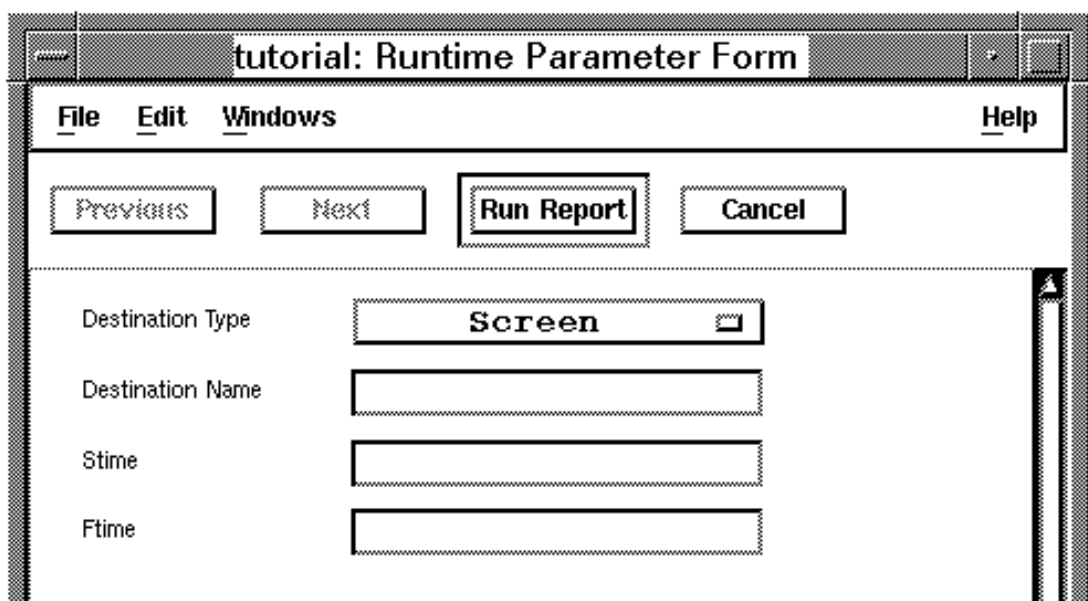


Figure 3-26. Runtime Parameter Form

This form lets you define any runtime (execution) parameters that you have in your report. In this case, you must enter values for STIME and FTIME which determine the time range for the query.

An example is shown in [Figure 3-27](#). You must substitute a time range that fits your system.

tutorial: Runtime Parameter Form

File Edit Windows Help

Previous Next Run Report Cancel

Destination Type Screen

Destination Name

Stime 09/05/96 23:30:00

Ftime 09/06/96 07:30:00

Figure 3-27. Specifying Values for Execution Parameters

22. Enter values for STIME and FTIME using [Figure 3-27](#) as a guide, and then click on **Run Report**.

This runs the report. After a few moments, the report is displayed in the Preview window, [Figure 3-28](#).

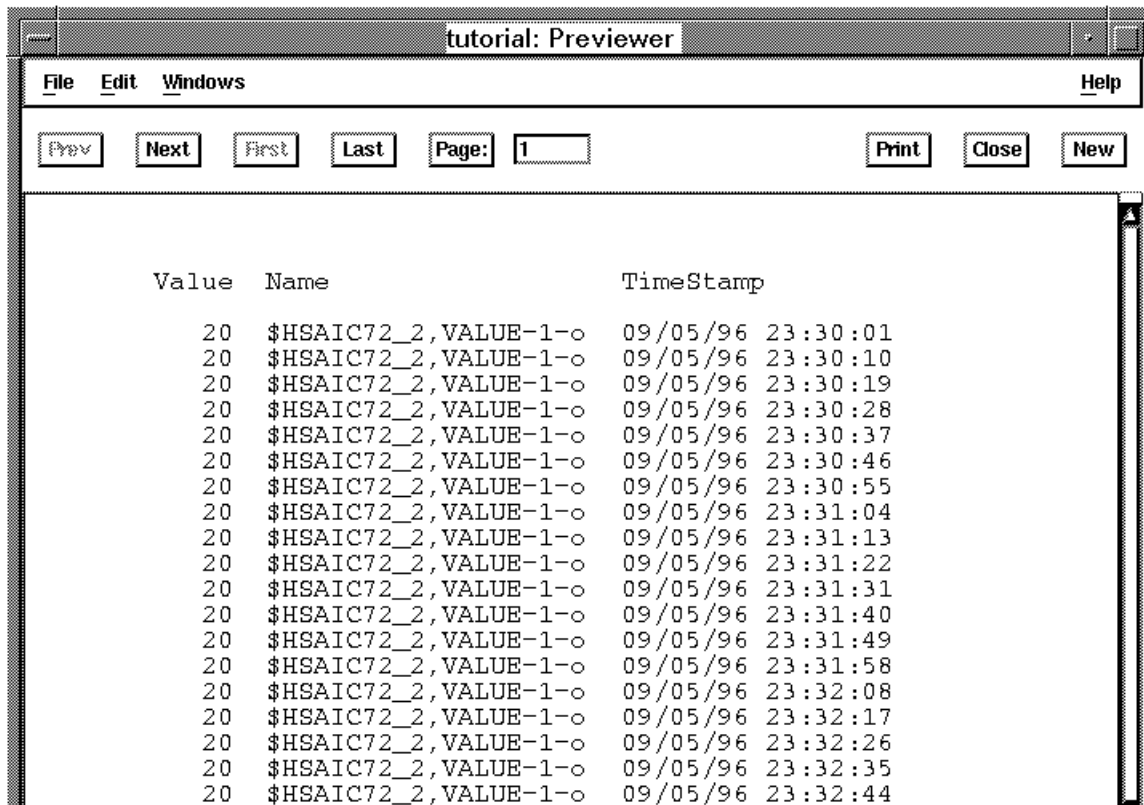


Figure 3-28. Preview Window

23. Click on **Close** to close the Preview window.

You are now finished building the report template in Oracle Reports.

24. To exit from the Oracle Reports application, choose **Quit** from the **File** menu in any of the Oracle Reports windows that are currently open (Object Navigator, Data Model, or Report Layout).

3.4.3 Create a New Report Object Instance

A report object instance is the combination of a report template (configured via Oracle Reports) and scheduling and execution parameters specified via the Report Configuration window. To create a new report object instance:

1. Choose **New** from the **File** menu in the Report Configuration window.

This displays the New Report dialog box, [Figure 3-29](#).

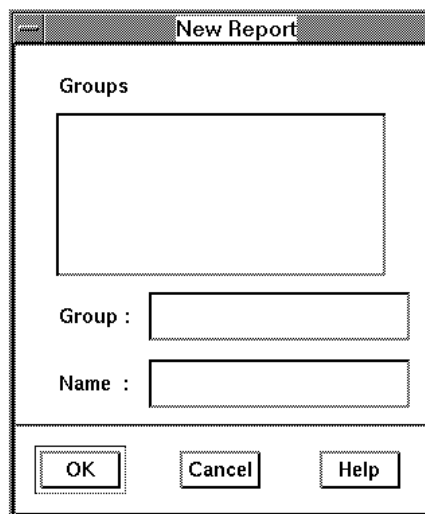


Figure 3-29. New Report Dialog Box

2. Enter a group name into the Group field and enter a report object instance name in the Name field, [Figure 3-30](#). The characters allowed are: a - z, A - Z, 0 - 9, dash (-), and underscore (_).

Group :

Name :

Figure 3-30. Entering the Report Object Instance Group and Name

This assigns the report to the group. The group is a means to organize related report object instances. Note that if your system has existing groups, they are displayed in the Groups selection box. You can select an existing group rather than define a new one.

3. Click on **OK** to accept the entries.

This removes the dialog box. The data you entered in the dialog are displayed in the applicable fields of the Report Configuration window, [Figure 3-31](#).

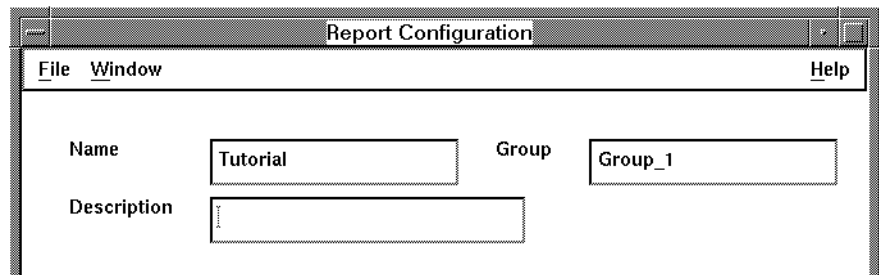


Figure 3-31. New Report Object Instance Name

4. Enter a description in the Description field. This step is optional.

3.4.3.1 Assign the Report Template to the Report Object Instance

To assign a report template to the report object instance, enter the report template name in the Template field, [Figure 3-32](#).



Figure 3-32. Assigning a Report Template to the Report Object Instance

This is the report template that you built in Oracle Reports as described in [Section 3.4.2, Build the Report Template](#). If you are not sure of the Template name, you can use the **Modify** button to display a list of existing templates.

3.4.3.2 Enable Scheduling Modes

There are three scheduling modes: Manual, Program, and System. You can enable one or any combination of these modes by selecting the corresponding button, [Figure 3-33](#).

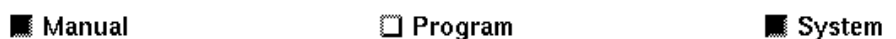


Figure 3-33. Enable Buttons for Scheduling Modes

Manual - allows scheduling by operator via the Manual Execution window.

Program - allows scheduling via commands from a program via AdvaInform User API, or a REPORT statement in a TCL program.

System - allows scheduling according to conditions specified via Condition List in the Report Configuration window (See [Section 3.4.3.4, Schedule the Report for Execution](#)).

3.4.3.3 Configure the Parameter List (Execution Parameters)

All reports have three standard execution parameters that can be used to specify the destination of the report output:

- The File parameter is used to specify an ASCII file where the report can be stored. Note that the report can not be retrieved from this file for future use in AdvaInform Reports.
- The Printer parameter is used to specify which printer will be used when the report is printed.
- The History parameter is used to specify the Report log (configured in AdvaInform History) where the report will be stored.

In addition you can specify values for execution parameters used in the SQL queries in the report template.

In this tutorial you will specify printer and history parameters (not the file parameter). In addition, you will specify values for the STIME and FTIME execution parameters used in the report template to define the time range for the query.

To configure the parameter list:

1. Click on the **New** button by the Parameter List box in the Report Configuration window, [Figure 3-34](#).

Parameter List

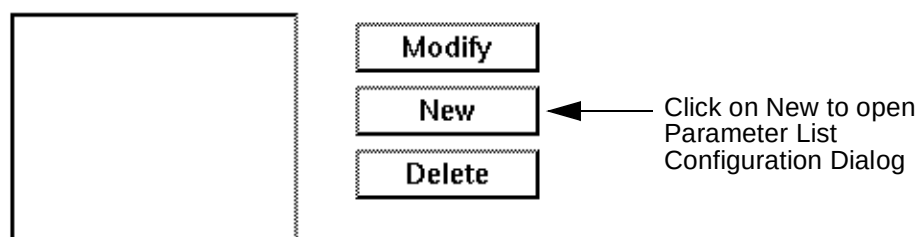


Figure 3-34. Parameter List Box

This displays the Parameter List Configuration Dialog Box, [Figure 3-35](#).

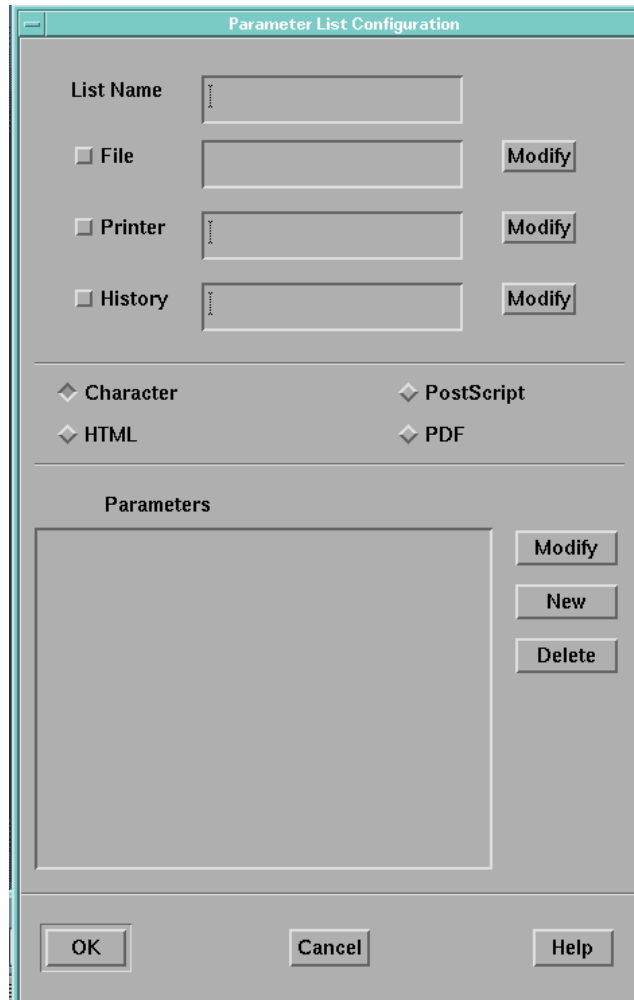


Figure 3-35. Parameter List Configuration Dialog Box

2. Enter a name for the parameter list in the List Name field, [Figure 3-36](#).



Figure 3-36. List Name

This name is used when you configure the condition list where you specify which parameter list to use when you execute the report.

3. In the Printer field specify where (on which printer) the report will be printed when it is executed. Click on the **Printer** button to enable the parameter, and then enter the printer name, [Figure 3-37](#).

The image shows a configuration interface for a printer. On the left, there is a checkbox labeled "Printer". To its right is a text input field containing the text "ReportPrinter". Further to the right is a button labeled "Modify".

Figure 3-37. Printer Name

If you do not know the name of the printer, you can click on the **Modify** button to display a list of printers to select from.

4. In the History field, specify a report log to store the report in History. Click on the **History** Button to enable the parameter, and then enter the name of the report log where you want to store the finished report, [Figure 3-38](#). The name is the one assigned to the log when it was configured via AdvaInform History.

The image shows a configuration interface for a report log. On the left, there is a checkbox labeled "History". To its right is a text input field containing the text "REPORT_LOG_1". Further to the right is a button labeled "Modify".

Figure 3-38. Report Log Name

5. Select an output mode for the report instance. There are four choices, [Figure 3-39](#).

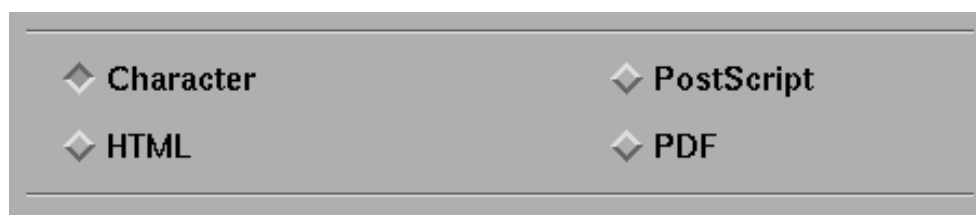
The image shows a configuration interface for output mode. It features four radio button options arranged in a 2x2 grid: "Character", "PostScript", "HTML", and "PDF".

Figure 3-39. Output Mode

For this tutorial, use Character mode. This way you can view the report on the screen without having to enter a command line to launch a third-party viewer. For an explanation of the other output modes, refer to [Section 3.6.2.1, Parameter List Configuration Dialog](#).

NOTE

If you select Character mode here, you must also specify Character mode when you build the report template. See [Section 3.4.2.2, Select the Output Mode for the Report](#).

6. Specify the STIME and FTIME execution parameters that define the start and end times for your query.
 - a. Click on the **New** Button by the Parameters box.

This displays the Parameter Edit dialog box, [Figure 3-40](#).

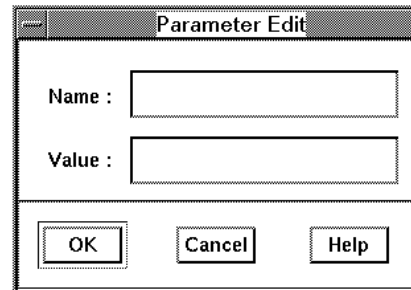
A screenshot of the 'Parameter Edit' dialog box. It has a title bar with a minus sign and the text 'Parameter Edit'. Inside, there are two text input fields. The first is labeled 'Name :' and the second is labeled 'Value :'. Below these fields are three buttons: 'OK', 'Cancel', and 'Help'.

Figure 3-40. Parameter Edit Dialog Box

- b. Enter the STIME parameter in the Name field, and enter the value in the Value field, [Figure 3-41](#).

Since the STIME value is a literal string with embedded spaces, it must be entered with single quotes. It is recommended that you use single quotes for any literal string that is passed to Oracle Reports.

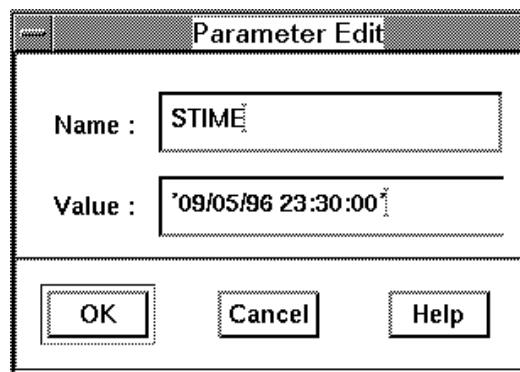
A screenshot of the 'Parameter Edit' dialog box, similar to Figure 3-40 but with data entered. The 'Name :' field contains the text 'STIME'. The 'Value :' field contains the text ''09/05/96 23:30:00''. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

Figure 3-41. STIME Definition

- c. Click on **OK** to accept the entries.

The STIME parameter definition is now displayed in the Parameters box in the Parameter List Configuration dialog box.
 - d. Repeat steps 6a, 6b, and 6c to specify the FTIME execution parameter (for example: **'09/06/96 07:30:00'**).

The Parameters box now contains the specified parameter values, [Figure 3-42](#).

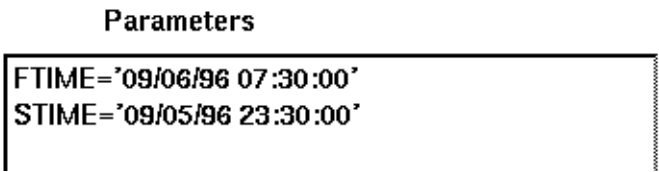


Figure 3-42. Completed Parameter List Configuration

- 7. Click on **OK** in the Parameter List Configuration dialog to accept the entries in the parameter list.

This closes the dialog box. The name of the new parameter list is entered in the Parameter List box in the Report Configuration window, [Figure 3-43](#).

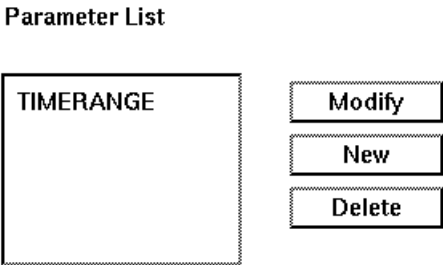


Figure 3-43. New Parameter List

3.4.3.4 Schedule the Report for Execution

You specify the schedule for the report via the Condition list, [Figure 3-44](#). In addition you select the appropriate parameter list to use when you execute the report.

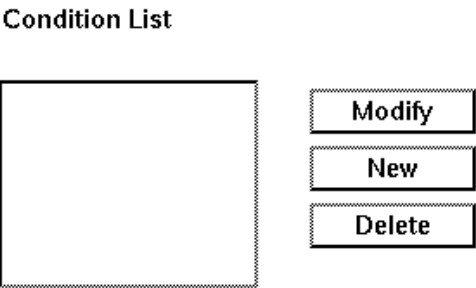


Figure 3-44. Condition List

For this tutorial define a condition that schedules the report each Friday morning at 9:30 using the TIMERANGE parameter list.

To specify the conditions:

1. For systems with Master software, click on the **New** button by the Condition Box. This displays the Execution Type Select box, [Figure 3-45](#). Then click on the **New Cyclic** button.

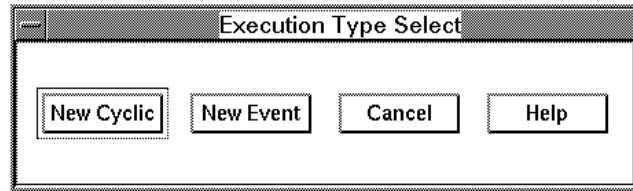


Figure 3-45. Execution Type Select dialog

For systems with MOD 300 software, click on the **New** button by the Condition Box.

Whether you clicked **New Cyclic** (for Master) or **New** (for MOD 300), the Execution Configuration dialog box is displayed, [Figure 3-46](#).

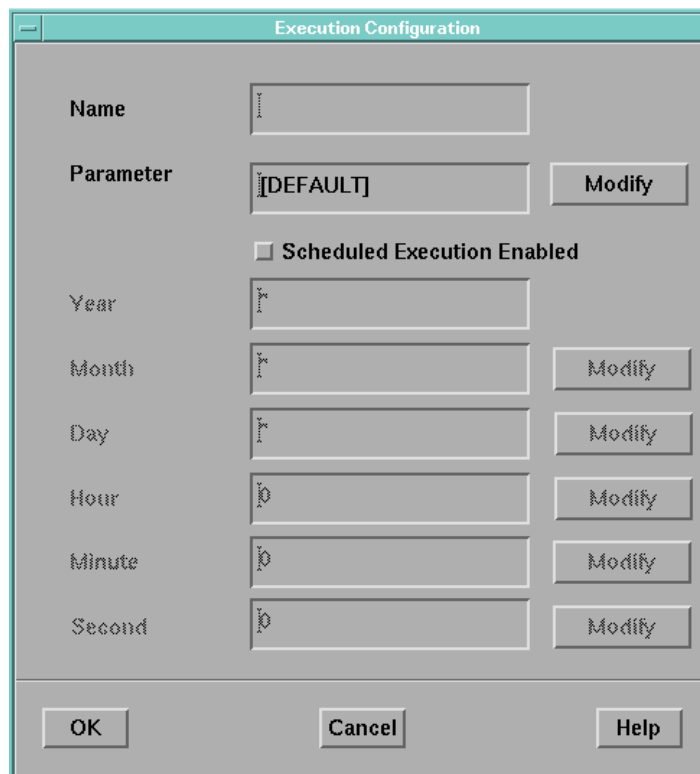


Figure 3-46. Execution Configuration Dialog Box

2. Enter the condition name in the Name field, [Figure 3-47](#).

Name

Figure 3-47. Condition List Name

3. Select the parameter list to be used when the report executes as follows:
 - a. Click on the **Modify** Button by the Parameter field.
This displays the Parameter Selection box, [Figure 3-48](#).

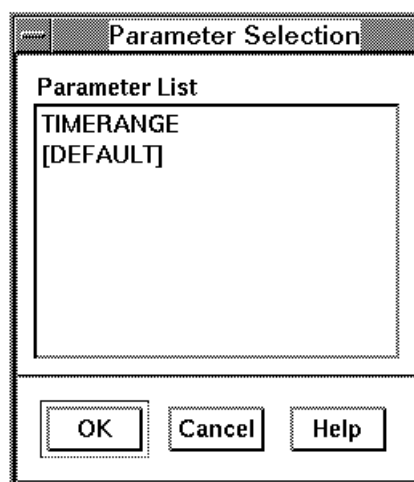


Figure 3-48. Parameter Selection

- b. Select the TIMERANGE parameter list and then click on **OK**.
This removes the selection box. The selected parameter list is displayed in the Parameter List field in the Execution Configuration dialog box, [Figure 3-49](#).

Parameter

Figure 3-49. Selected Parameter List

4. Enable Cyclic Execution by selecting the Scheduled Execution Enabled target, [Figure 3-50](#).

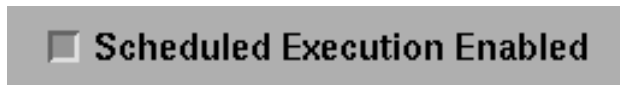


Figure 3-50. Cyclic execution Enabled Target

This means the report will be executed according to the schedule you specify in this dialog box. If cyclic execution is not enabled, the report will not execute according to the specified schedule in this dialog box.

5. Configure the Year, Month, Day, Hour, and Minute fields to schedule the report to execute every Friday morning at 9:30.

The year and the month fields are at their respective default values (all years and all months as indicated by the asterisk). Leave these fields at their defaults.

- a. Click on the **Modify** Button by the Day field.

This displays the Day Configuration dialog box, [Figure 3-51](#).

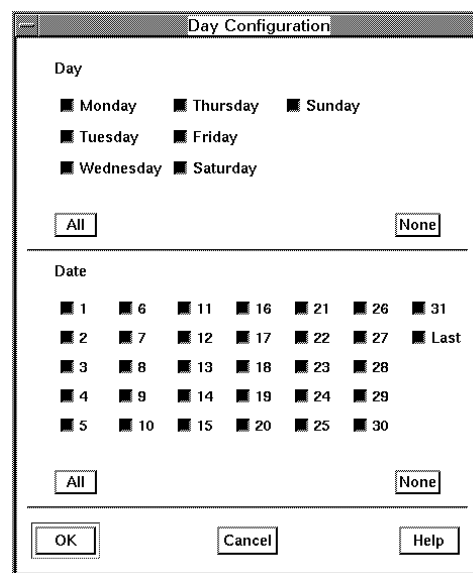


Figure 3-51. Day Configuration Dialog Box

This dialog box has two parts. On top you select days of the week, and on the bottom you select days of the month. The top and bottom parts work in combination to specify the days of the month when the report will execute. Initially, all the days are selected.

- b. To select just Friday, first click on the **None** button in the top part to de-select all days of the week, select **Friday**, [Figure 3-52](#), and then click on **OK**.

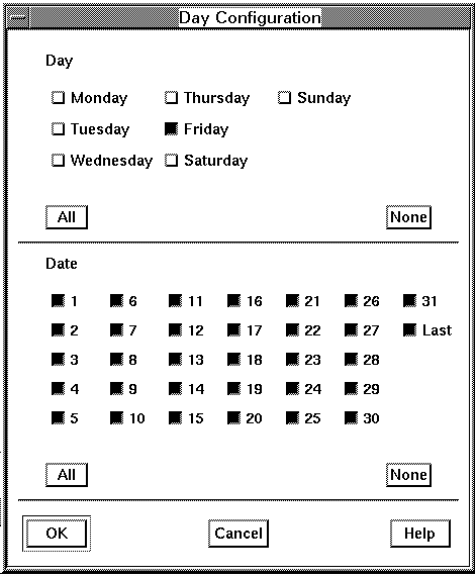


Figure 3-52. Day Configuration Dialog Set for Every Friday of the Month

The day configuration is entered in the Day field of the Execution Configuration Window, [Figure 3-53](#).

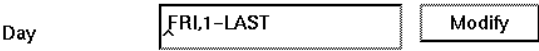


Figure 3-53. Day Configuration Entered in Day Field

FRI, 1 - LAST means every Friday from the first day of the month through the last.

- c. Click on the **Modify** Button by the Hour field.

This displays the Hour Configuration dialog box, [Figure 3-54](#).

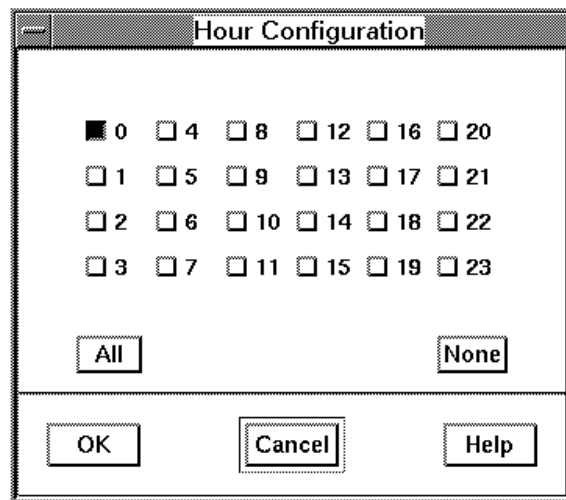


Figure 3-54. Hour Configuration Dialog Box

- d. Remove the selection from 0 by clicking on 0 (or **None**). Then select **9**, [Figure 3-55](#), and click on **OK**. This selects 9:00 AM.

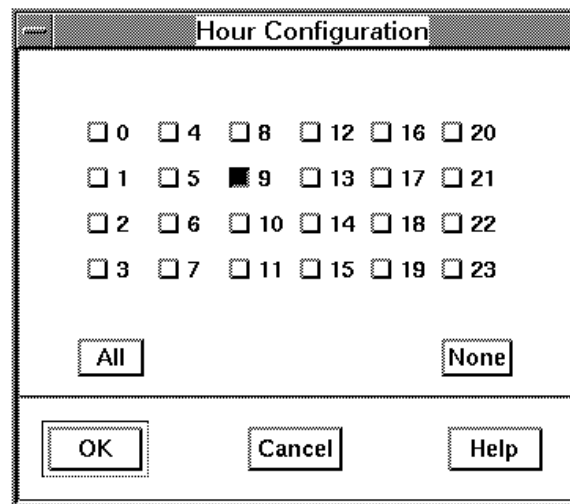


Figure 3-55. Hour Configuration for 9 AM

- e. Click on the **Modify** Button by the Minute Field.

This displays the Minute Configuration dialog box, [Figure 3-56](#).

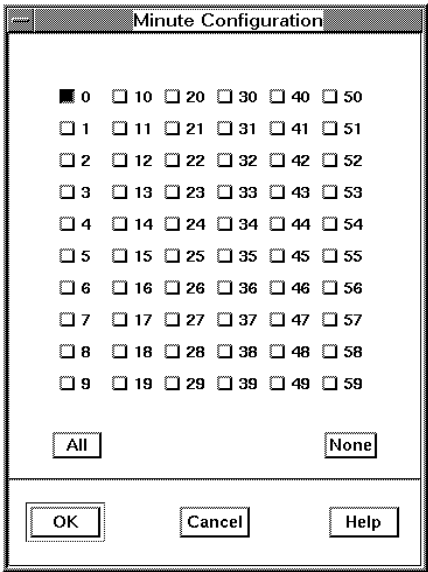


Figure 3-56. Minute Configuration Dialog Box

- f. Remove the selection from 0 by clicking on its box (or **None**). Then select **30**, [Figure 3-57](#), and click on **OK**. This selects 30 minutes after the selected hour.

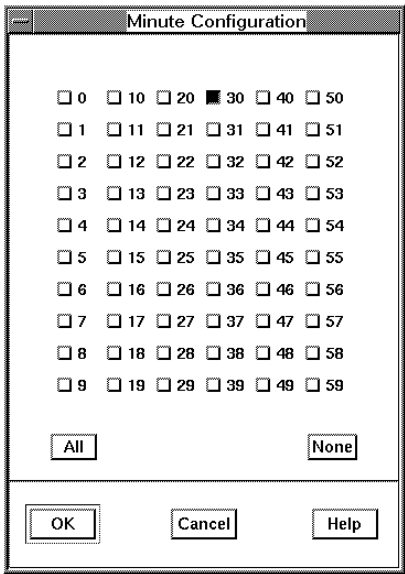


Figure 3-57. Minutes Configuration for 30 Minutes After the Hour

The completed condition is shown in [Figure 3-58](#).

The image shows a dialog box titled "Execution Configuration". It contains the following fields and controls:

- Name:** A text box containing "FRIDAY_REPORT".
- Parameter:** A text box containing "TIMERANGE" with a "Modify" button to its right.
- Scheduled Execution Enabled:** A checkbox that is currently unchecked.
- Year:** A text box with a calendar icon.
- Month:** A text box with a calendar icon and a "Modify" button to its right.
- Day:** A text box containing "FRI,1-LAST" with a "Modify" button to its right.
- Hour:** A text box containing "8" with a "Modify" button to its right.
- Minute:** A text box containing "30" with a "Modify" button to its right.
- Second:** A text box containing "0" with a "Modify" button to its right.
- Buttons:** At the bottom are "OK", "Cancel", and "Help" buttons.

Figure 3-58. Completed Execution Configuration (Condition List)

- Click on **OK** to accept the entries in the condition list.

The condition list is now displayed in the Report Configuration window, [Figure 3-59](#).

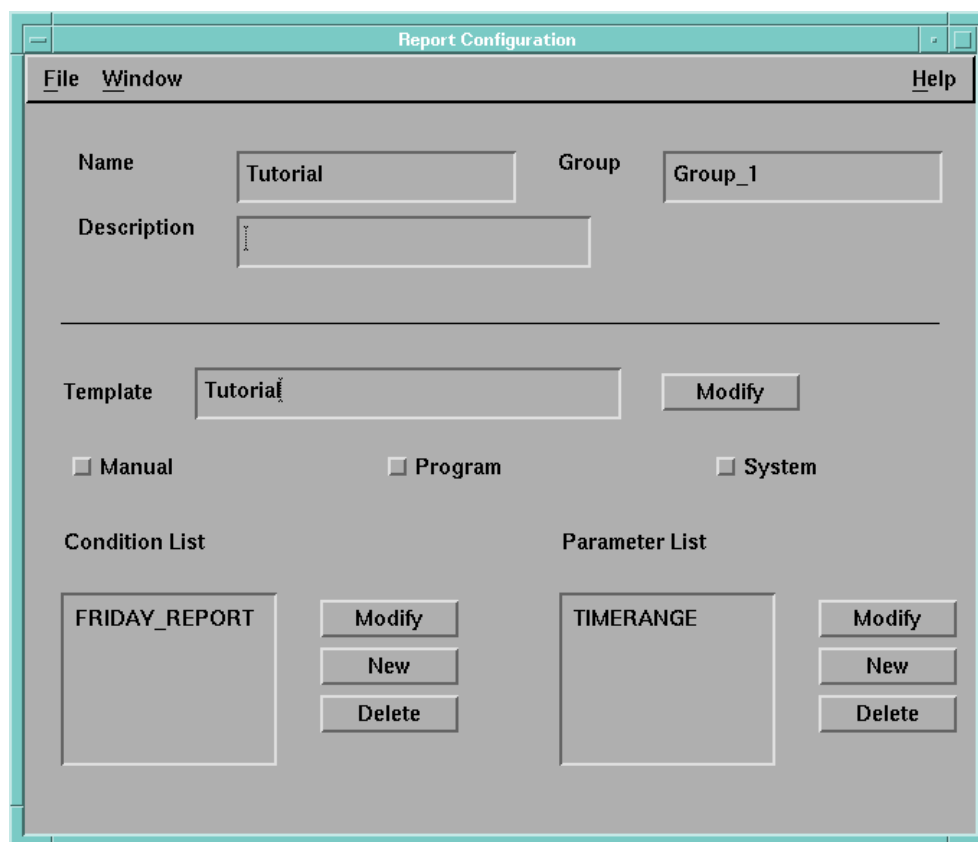


Figure 3-59. Completed Report Object Instance Configuration

7. To save the configuration choose **Save** from the **File** menu.

This completes the report configuration. The report executes each Friday at 9:30 and it can also be scheduled manually.

NOTE

This report is now scheduled to execute cyclically, week after week. If you do not want to keep the report you can delete the report object instance using the **Delete** menu item in the **File** menu of the Report Configuration window. Refer to [Section 3.6.2.12, Report Configuration Window - File Menu](#) for details.

3.5 Application Procedures

The following is a summary of application procedures for building a report.

- [Section 3.5.1, How to Open the Report Configuration Window](#)
- [Section 3.5.2, How to Build Report Templates with Oracle Reports](#)
 - [Section 3.5.2.1, How to Open an Existing Report](#)
 - [Section 3.5.2.2, How to Make Data Available to Other Reports \(Encapsulate Data\)](#)
 - [Section 3.5.2.3, How to Use Execution Parameters in a Query](#)
- [Section 3.5.3, How to Create a Report Object Instance](#)
- [Section 3.5.4, How to Back Up and Restore Report Templates](#)
- [Section 3.5.5, How to Integrate Graphic Charts in a Report](#)
- [Section 3.5.6, How to Send Report Output to a File](#)
- [Section 3.5.7, How to Print a Report](#)
- [Section 3.5.8, How to Store Report Output in History](#)

3.5.1 How to Open the Report Configuration Window

To open the Report Configuration window, first choose **Reports** from the **AdvaInform** menu on the IMS Menu, then choose **Report Configuration** from the **Reports** sub-menu. For details on the Report Configuration window, refer to [Section 3.6.2, Report Configuration Window](#).

3.5.2 How to Build Report Templates with Oracle Reports

You build report templates with Oracle Reports. This book does not describe Oracle Reports in detail. Rather, it describes how it is applied in AdvaInform Reports. For information regarding Oracle Reports, see the Oracle Reports books listed in [Section 1.5, Related Documentation](#).

To start building a new report, first open the Oracle Reports Object navigator by choosing **Build Reports** from the **Windows** menu in the Report Configuration window. Then double-click on the Data Model icon under the UNTITLED report to start a new report template. An example of how to build a report template is provided in [Section 3.4.2, Build the Report Template](#). For further details regarding operation of the Report Configuration window, refer to [Section 3.6.2, Report Configuration Window](#).

The following three sections provide instructions for procedures related to building a report template that are not covered in the tutorial. They are:

- [Section 3.5.2.1, How to Open an Existing Report](#)
- [Section 3.5.2.2, How to Make Data Available to Other Reports \(Encapsulate Data\)](#)
- [Section 3.5.2.3, How to Use Execution Parameters in a Query](#)

[Section 3.7, How to Write Queries for Reports](#) provides a number of example queries for common report applications such as: process data, history message logs, and numeric history logs. Use the example queries there as guidelines for building your own queries.

3.5.2.1 How to Open an Existing Report

The report templates you build with Oracle Reports are stored in the following directory:

`/home/opt/advant/Reports/customer/templates`

The report template files have the following extension: `.rdf`

To open an existing report:

1. Choose **Open** from the **File** menu.

This displays a dialog for navigating directories and files in the Oracle database, [Figure 3-60](#). The default directory is the one where your reports are stored.

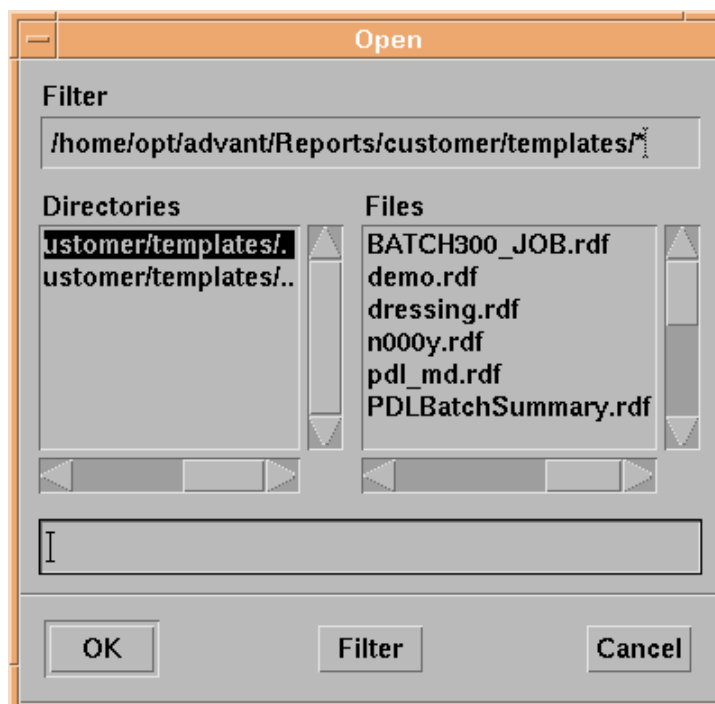


Figure 3-60. Open Dialog Continued

2. Use the Open dialog to select the report you want to open and then click on **OK**.

The selected report and related objects are listed in the Object Navigator window, [Figure 3-61](#).

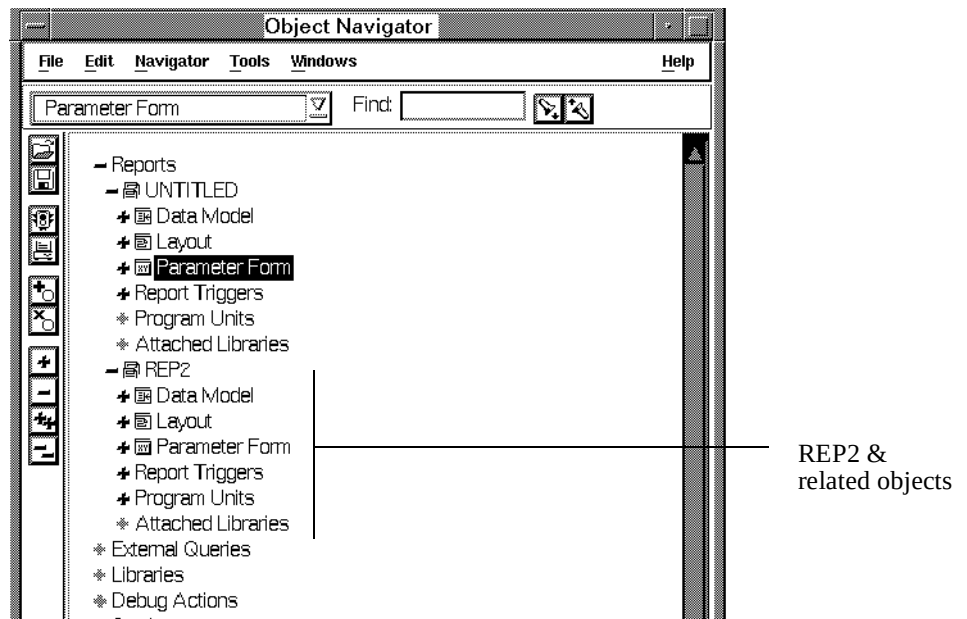


Figure 3-61. Selected Report Presented in Object Navigator Window

3. To open a view of the report, double-click on the appropriate object under the report name.

The Data Model view is where you generally begin report building. This is where you write the SQL queries to retrieve data for the report.

3.5.2.2 How to Make Data Available to Other Reports (Encapsulate Data)

You can encapsulate the data for any field in the report such that it can be accessed by another SQL query on an individual basis (independent of the rest of the report). The query can either be part of another report, or it can be invoked using the Database Access function in AdvaInform Basic Functions. This is done via the user_exit *strhis*.

A user_exit is a subroutine linked into Oracle Reports executables. When Oracle Reports calls this routine, control temporarily passes to the user_exit. When the user_exit finishes executing, control passes back to Oracle Reports.

The *strhis* user_exit can only be used on reports that are to be stored in a report log. Data is encapsulated on a field basis. When a field is encapsulated via *strhis*, its value is sent to the same report log that stores the finished report. Other report templates can use SQL queries to retrieve the encapsulated data from the report log.

To encapsulate a field on a report:

1. Open the Data Model view of the report whose field you want to encapsulate, [Figure 3-62](#).

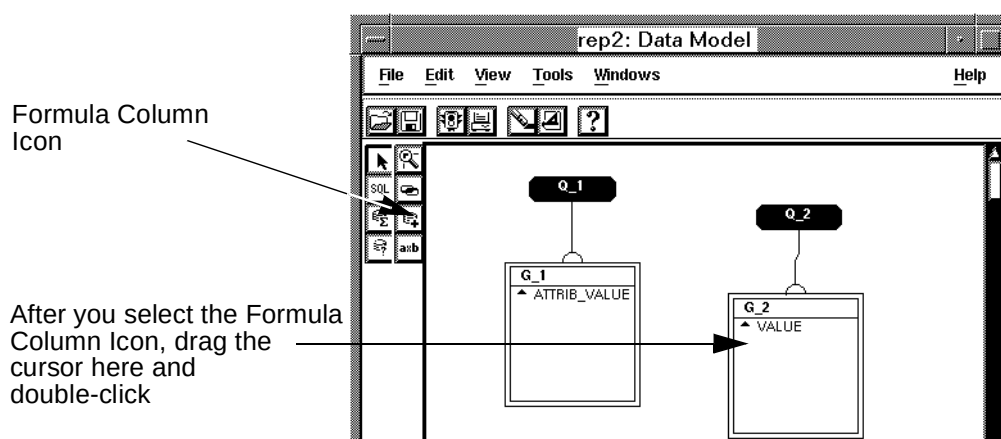


Figure 3-62. Data Model View of Selected Report

2. Double-click on the Formula-Column icon, [Figure 3-62](#).
This converts the cursor to a cross-hair.
3. Drag the cursor to the query that contains the field you want to encapsulate and double-click.

This displays the Formula Column dialog box, [Figure 3-63](#).

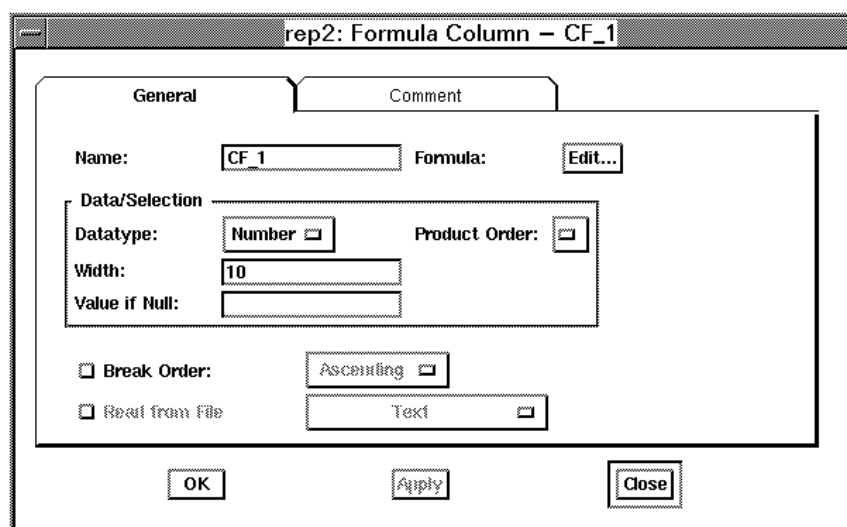


Figure 3-63. Formula Column Dialog Box

4. Click on the **Edit** button.

This displays a dialog for entering the User_Exit script, [Figure 3-64](#).

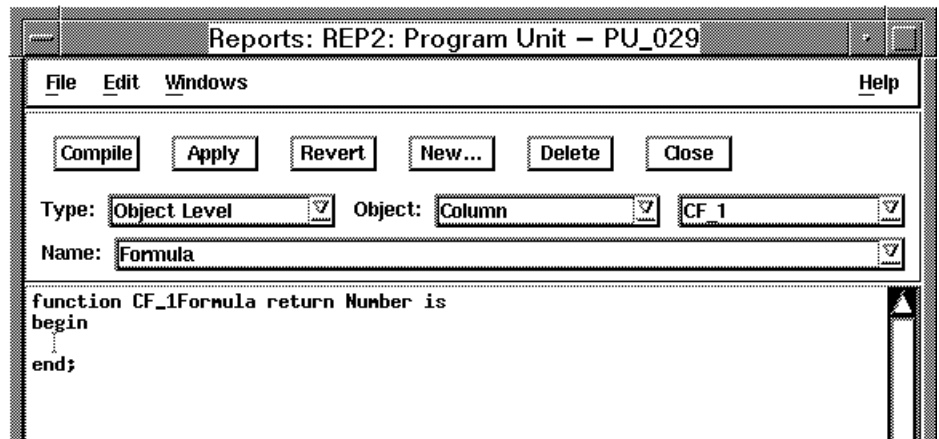


Figure 3-64. Program Unit Dialog

The starting (begin) and ending (end;) phrases are already entered for you.

5. Finish the script by entering the following lines between `begin` and `end;`.

```
srw.reference (:columnname);
srw.user_exit ('strhis columnname');
```

where `columnname` = the name of the column specified in the select statement that identifies the database attribute. An example is shown in [Figure 3-65](#).

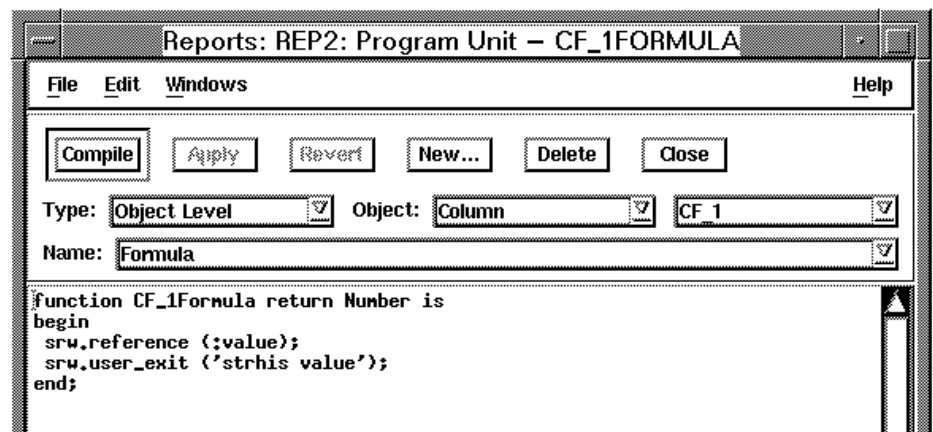


Figure 3-65. Program Unit Dialog, Continued

6. Click on the **Compile** button when you are finished with the script, and then click on **Close** to close the window.

This returns you to the Formula Column window.

7. Click on **Close** to close this window also.

The existence of an encapsulated field is indicated by CF_n in the query box where n represents the sequential number of the encapsulated field, [Figure 3-66](#).

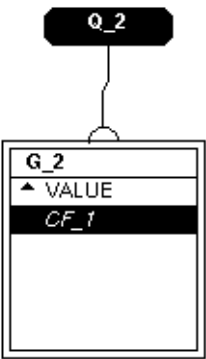


Figure 3-66. Indication of Encapsulated Field

When you specify a field for encapsulation, each instance of that field results in an instance of stored data in the report log. For example, if the query that sets up the MEASURE3 (MOD 300) or VALUE (Master) field in the previous example retrieves three instances of MEASURE3 (or VALUE), then there will be three instances of encapsulated data stored in the report log, [Figure 3-67](#)

Wkly_Boiler_15001					
File	Action	Window	Help		
Report	User Name	Original Time	Stored Time	Archived	Size
VALUE	Jim	17-Sep-1993 12:30:52	17-Sep-1993 12:30:52	Yes	251
[VALUE	Jim	im24-Sep-1993 12:30:05	24-Sep-1993 12:30:05	Yes	251
VALUE	Jim	01-Oct-1993 12:31:18	01-Oct-1993 12:31:18	Yes	251
Boiler_15001	Jim	08-Oct-1993 12:30:27	08-Oct-1993 12:30:27	No	251
Boiler_15001	Jim	08-Oct 1993 12:32:10	08-Oct-1993 12:32:10	Yes	251
Report	User Name	Original Time	Stored Time		
Boiler_15001	Jim	08-Oct 1993 12:32:10	08-Oct 1993 12:32:10		

Figure 3-67. Report Log Showing Encapsulated Data Entries

3.5.2.3 How to Use Execution Parameters in a Query

An **execution parameter** is a place holder for variables in SQL queries in a report template. The actual data for the variable is specified in a parameter list that you configure when you create the report object instance as described in [Section 3.6.2.1, Parameter List Configuration Dialog](#).

Execution parameters let you change parameter values from one execution to the next. For example, in a MOD 300 application you may want to hold a place for the batch name as shown in the example below:

```
SELECT Product_Name, Batch_Goal, Batch_Actual
FROM job
WHERE batchid=:BATCH
```

In a Master application, you may want to hold a place for the object name as shown in the example below:

```
SELECT Name, Value, Unit
FROM AI
WHERE Name = :object1
```

Execution parameters are marked by the colon (:). For instance, in the MOD 300 example above, :BATCH is an execution parameter. In the Master example above, :object1 is an execution parameter.

Before you execute a report, you specify the actual values in parameter lists associated with the report object instance. For example, when the batch with identifier JOB3_DEC15_BTCH16 runs, you enter that batch identifier for the BATCH execution parameter before you run the batch report.

3.5.3 How to Create a Report Object Instance

The report object instance is created by mapping scheduling and execution parameters to a report template. This is done via the Report Configuration window. An example of how to create a report object instance is provided in [Section 3.4.3, Create a New Report Object Instance](#). For further details regarding operation of the Report Configuration window, refer to [Section 3.6.2, Report Configuration Window](#).

3.5.4 How to Back Up and Restore Report Templates

For Oracle Reports version 2.5, report templates are stored as files in the directory:

```
/home/opt/advant/Reports/customer/templates/
```

with the extension: .rdf. Backup copies can be made using the Unix tape archiving program: tar. To use tar:

1. Insert a backup tape in the default tape drive.
2. Enter the following command from a terminal window (hpterm or dtterm):

```
tar cv /home/opt/advant/Reports/customer/templates/*.rdf
```

3.5.5 How to Integrate Graphic Charts in a Report

You can integrate bar or pie charts, or other graphics in a report using the Graphic Builder function. This involves building and integrating a report template and a graphic template.

This section provides two quick tutorials that demonstrate how to integrate graphics in a report. These tutorials do not cover all steps in the same detail as in [Section 3.4, Tutorial](#) since you should already be familiar with most of the procedures. Any procedures unique to building graphic templates are covered in detail. If you need help on a report building procedure, refer to back to [Section 3.4, Tutorial](#).

The queries in these tutorials get their data from a sample table in the Oracle database. Before you begin, run the following Unix shell script to create the sample table:

```
/opt/advant/Reports/etc/OG_Example
```

3.5.5.1 Tutorial 1 - Basic Bar Chart

This tutorial demonstrates how to integrate a bar chart in a report. The query selects two rows (unitname and tonnage) from the sample table (example1). The data from one row (unitname) will be shown on the X axis, and the data from the other row (tonnage) will be shown on the Y axis. To build this graphic report:

1. Build a report template that retrieves the unitnames and tonnage for all units in the example1 table. Open the Oracle Reports Object Navigator by choosing **Build Templates** from the **Windows** menu in the Report Configuration window. Then create a query in the Data Model. The query is shown entered in the Query Properties Sheet, [Figure 3-68](#).

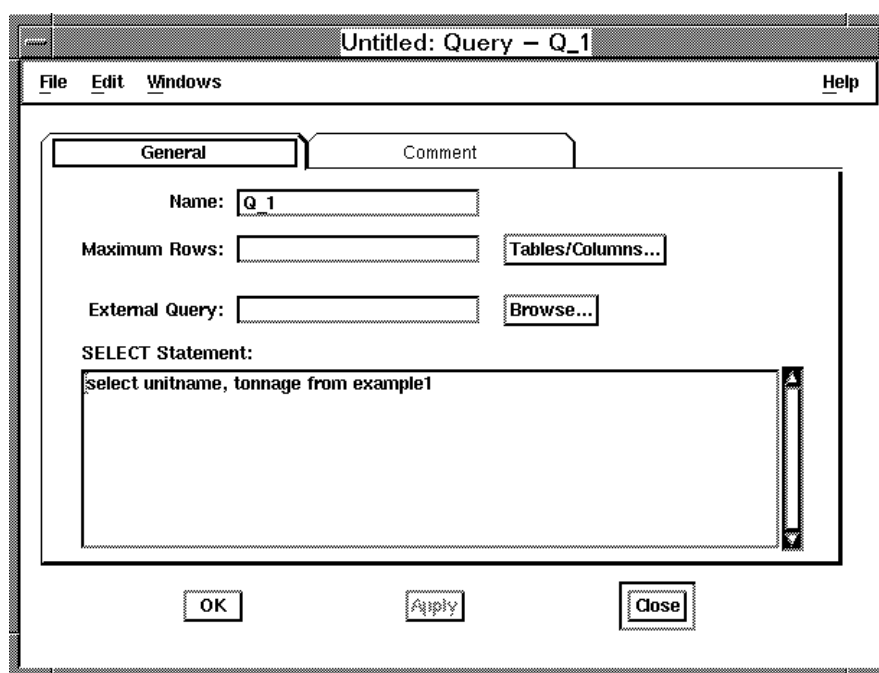


Figure 3-68. Sample Query

2. Create a default layout for the report. Choose **Default Layout** from the **Tools** menu in the Data Model window, and then select the default tabular layout in the Default Layout dialog.
3. Choose **Build Graphics** from the **Window** menu of the Report Configuration window. This opens the Oracle Graphics Object Navigator and Graphics Layout Editor.
4. Click on the Graphics icon in the Graphics Layout Editor, [Figure 3-69](#). Then drag the cursor to specify the location and size of the chart.

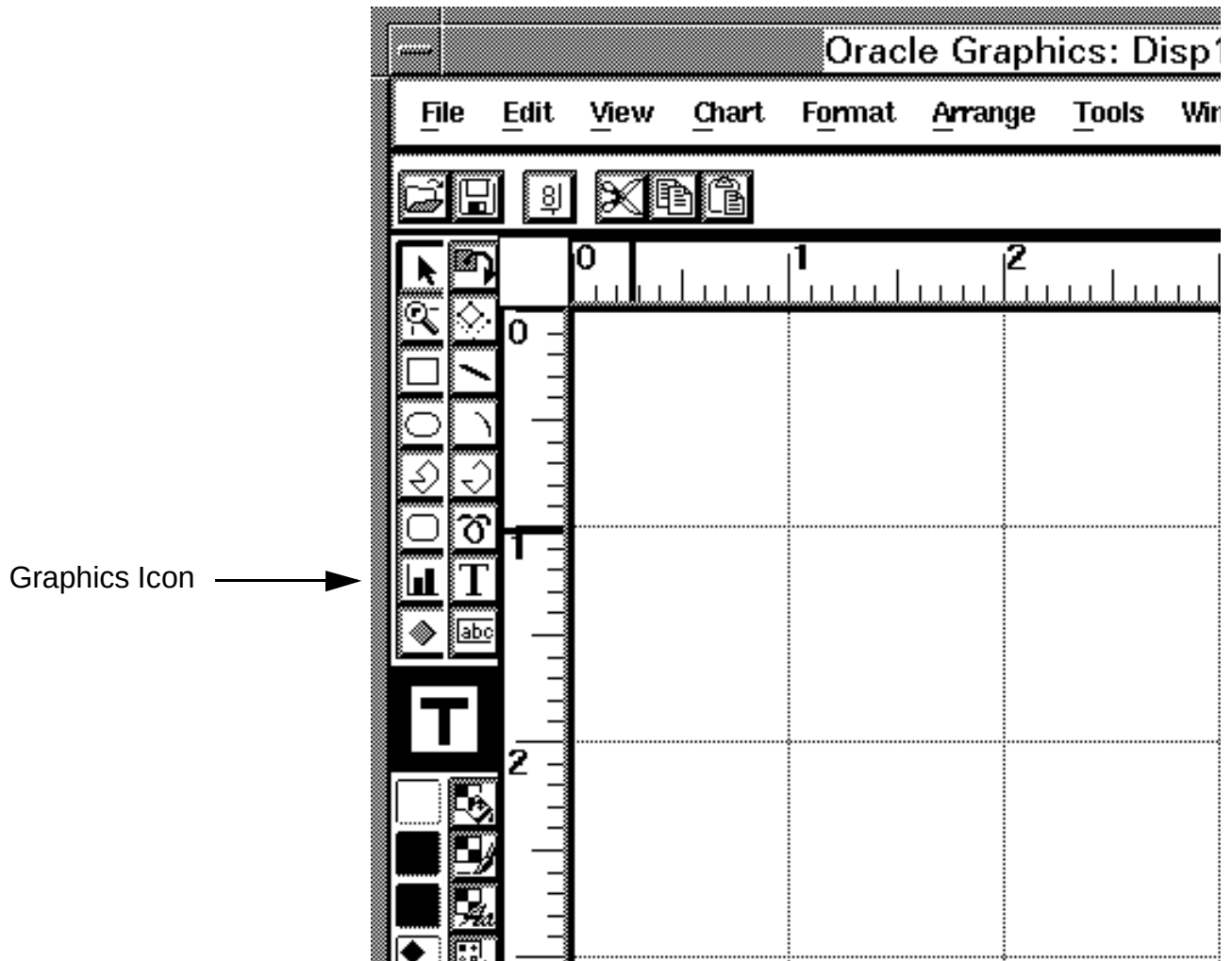


Figure 3-69. Graphics Layout Editor, Graphics Icon

When you release the mouse button, after a brief moment the Chart Genie dialog will be displayed, [Figure 3-70](#).

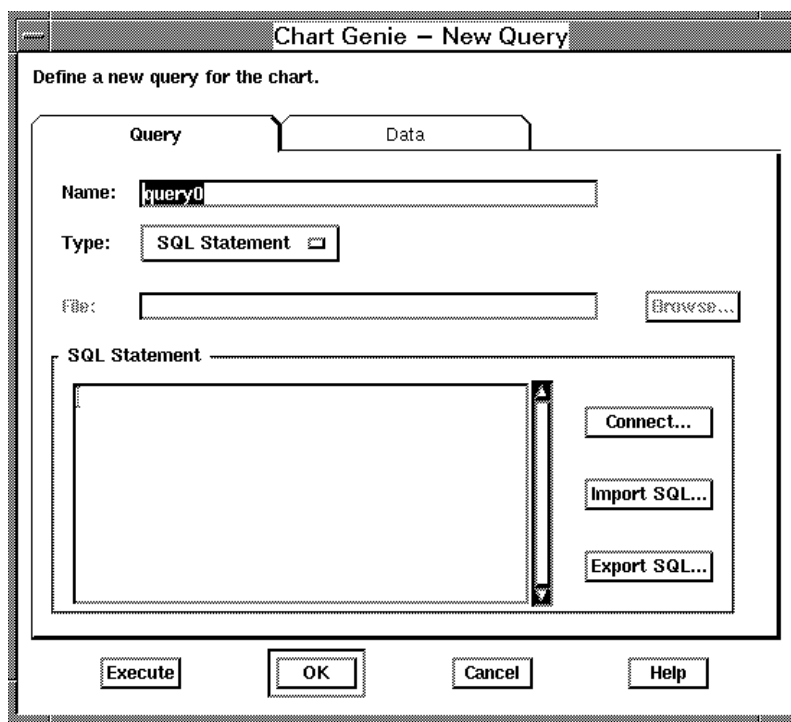


Figure 3-70. Chart Genie for New Query

5. Enter the same query as you entered for the report template, [Figure 3-71](#). You can keep the default query name (query0) or specify another name. Click **OK** when you finish.

```
select unitname, tonnage from example1
```

Figure 3-71. Entering Same Query for Graphic Template

When the Chart Genie dialog is removed, the Chart Properties dialog is displayed, [Figure 3-72](#).

6. Enter a name, and select a chart type. For this tutorial, select the bar type.

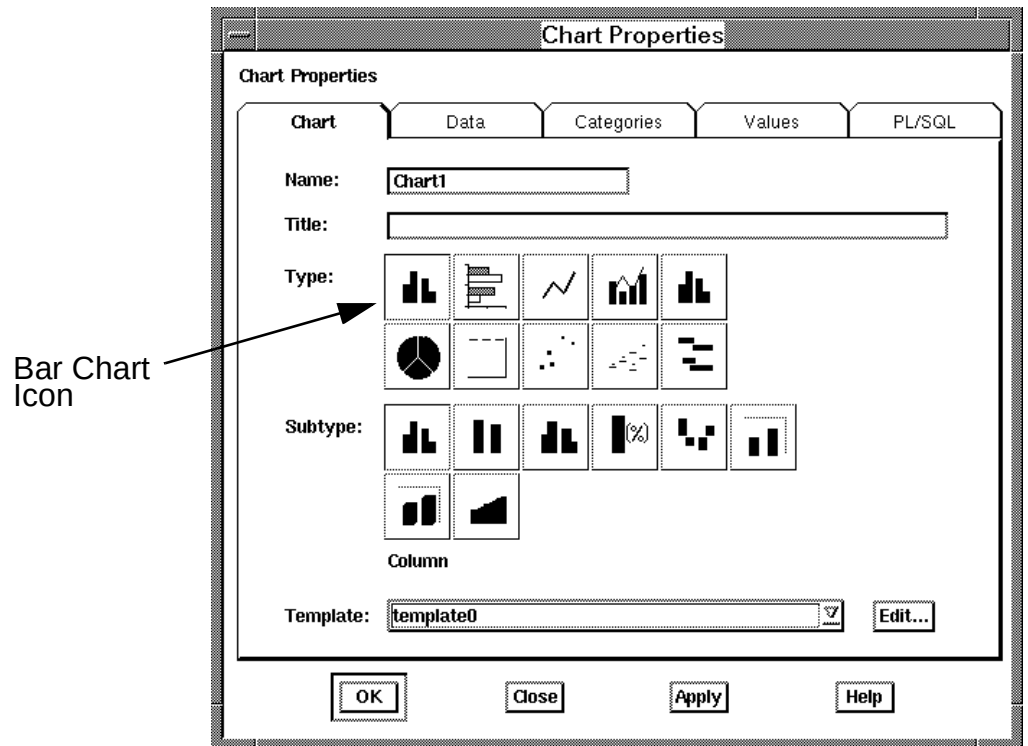


Figure 3-72. Chart Properties Dialog

When you finish with the Chart Properties Dialog, and the dialog box is removed, the bar chart will be displayed in the Graphics Layout Editor, [Figure 3-73](#).

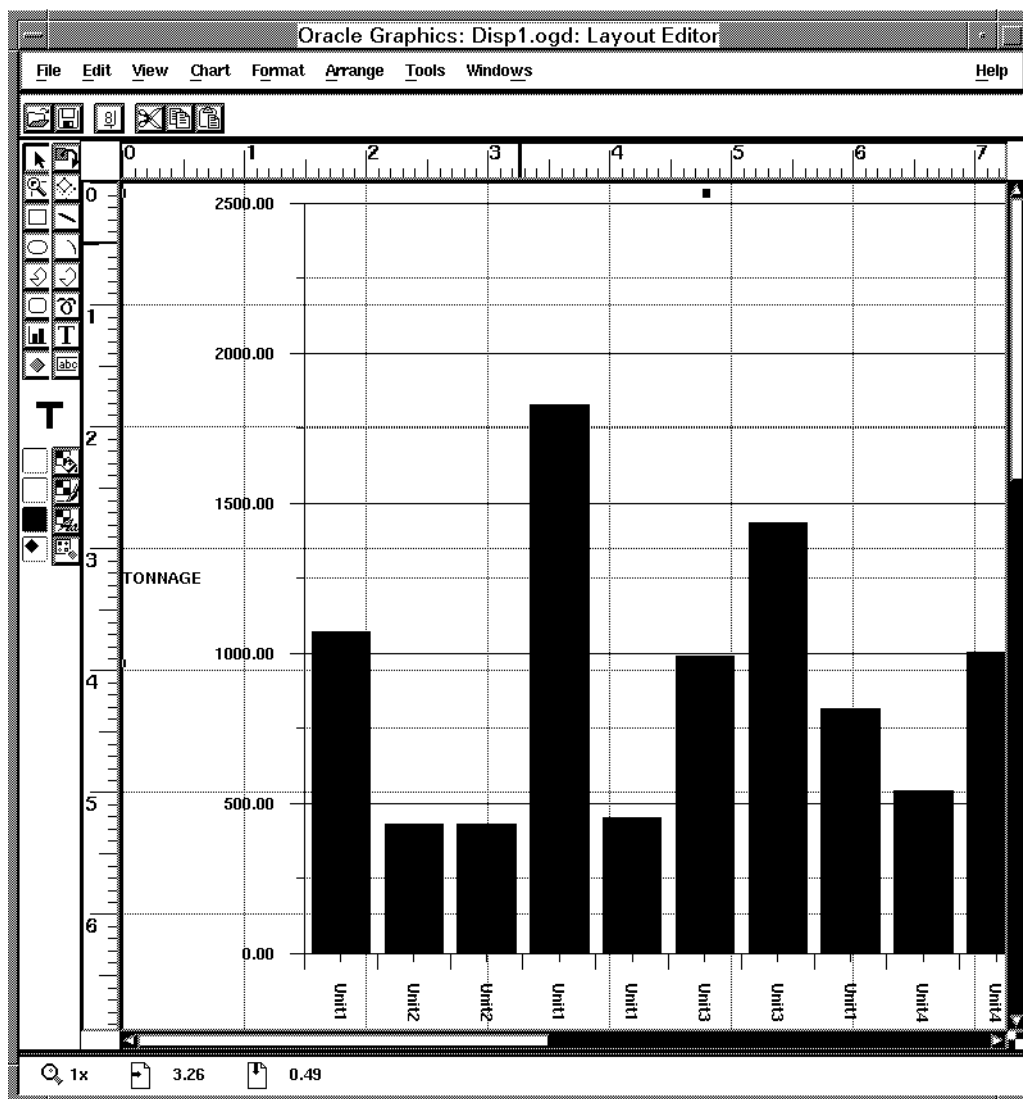


Figure 3-73. Bar Chart Shown in Graphic Layout Editor

7. Save the Graphic template to a file.

Choose **Save** from the **File** menu in the Graphics Layout Editor. In the save dialog, select the Save to File System option. In the Save File dialog, enter the path below:

`/home/opt/advant/Reports/customer/templates/filename.ogd`

where *filename* is the name you give the graphic. The default is Disp1. Graphic templates require the file extension .ogd.

8. Return to the Layout Editor for the Report Template you built in step 1.
9. Click on the Graphic icon, [Figure 3-74](#), and outline the area in the editor where you want to display the chart.

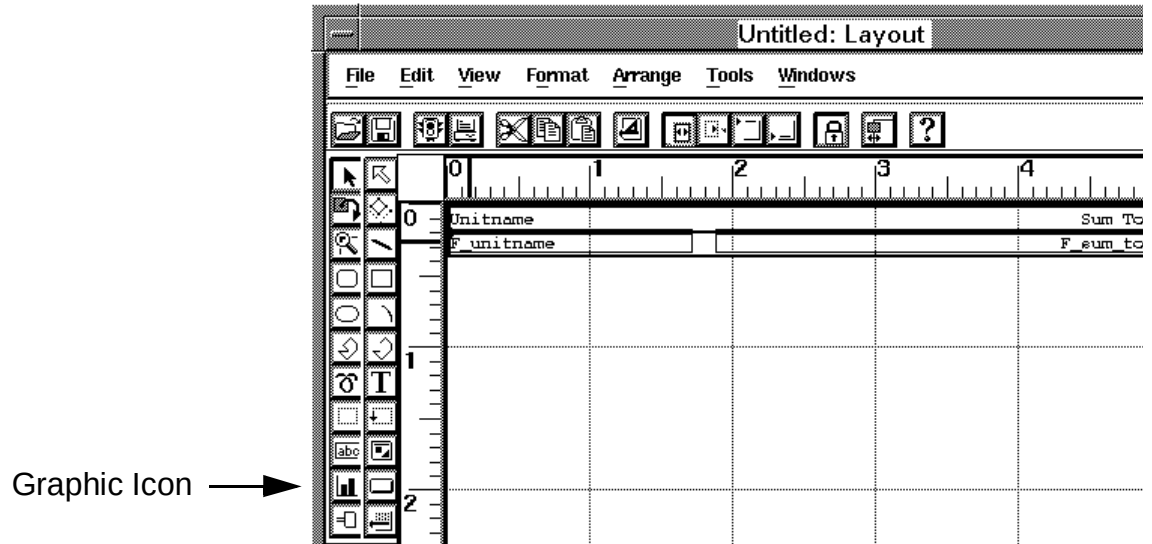


Figure 3-74. Graphic Icon in Report Layout Editor

This displays the Oracle Graphics Display dialog, [Figure 3-75](#).

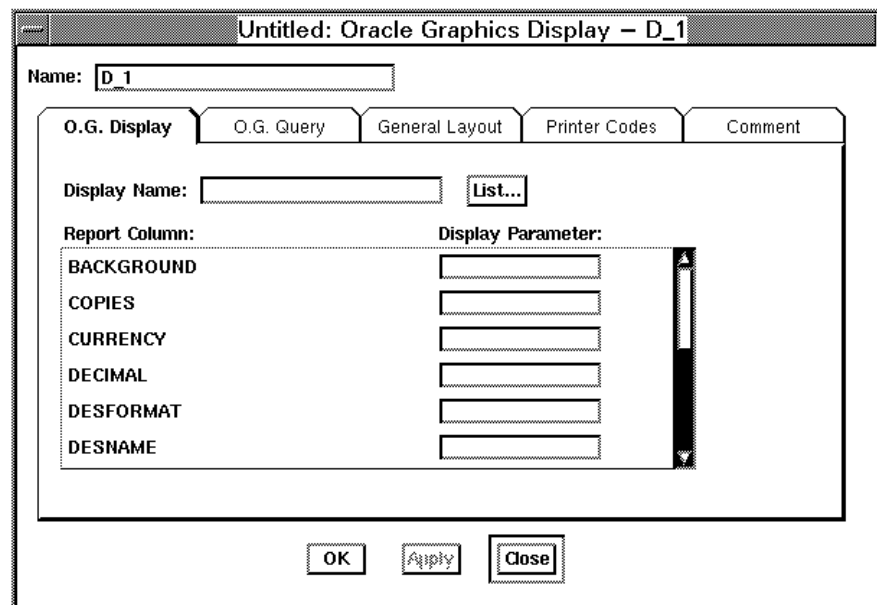


Figure 3-75. Oracle Graphics Display Dialog

This dialog is used to specify the data source for the graphic. Even though you created a query for the Graphic template, the actual data source is the query in the Report Template.

10. Enter the full path for the display name in the Display Name field. For example:

/home/opt/advant/Reports/customer/templates/Disp1.ogd

You can either enter the path and display name directly, or use the **List** button to display a dialog for selecting the appropriate path and display name.

11. Click on the **O.G. Query** tab. This displays the Query page of this dialog, [Figure 3-76](#).
12. In the Display Query field, enter the name of the query from the Graphic template (query0).
13. Use the button next to the Source field to select G_1.

G_1 is the Group for the query you created in the Report Template Data Model.

14. Scroll down the Report Column window to find the column names you specified in your query (tonnage and unitname), and select them here.

The screenshot shows a dialog box titled "Untitled: Oracle Graphics Display - D_1". It has five tabs: "O.G. Display", "O.G. Query" (which is selected), "General Layout", "Printer Codes", and "Comment".

Inside the "O.G. Query" tab, there are several fields and a list:

- Name:** A text field containing "D_1".
- Display Query:** A text field containing "query0".
- Source:** A dropdown menu showing "G_1" with a small square button to its right.
- Report Column:** A list box containing the following items: "MODE", "ORIENTATION", "PRINTJOB", "THOUSANDS", "tonnage", and "unitname". The "tonnage" and "unitname" items are highlighted in black.
- Display Column:** A list of empty text input fields, one for each item in the "Report Column" list.

At the bottom of the dialog, there are three buttons: "OK", "Apply", and "Close".

Figure 3-76. Oracle Graphic Display, O.G. Query Page

15. Click **OK** when you are finished.
16. To test the report, choose **Run** from the **File** menu, and then press **Run** report in the Runtime Parameter form. This generates a preview of the report, [Figure 3-77](#).
17. Create a Report Object Instance using the Report Configuration window.

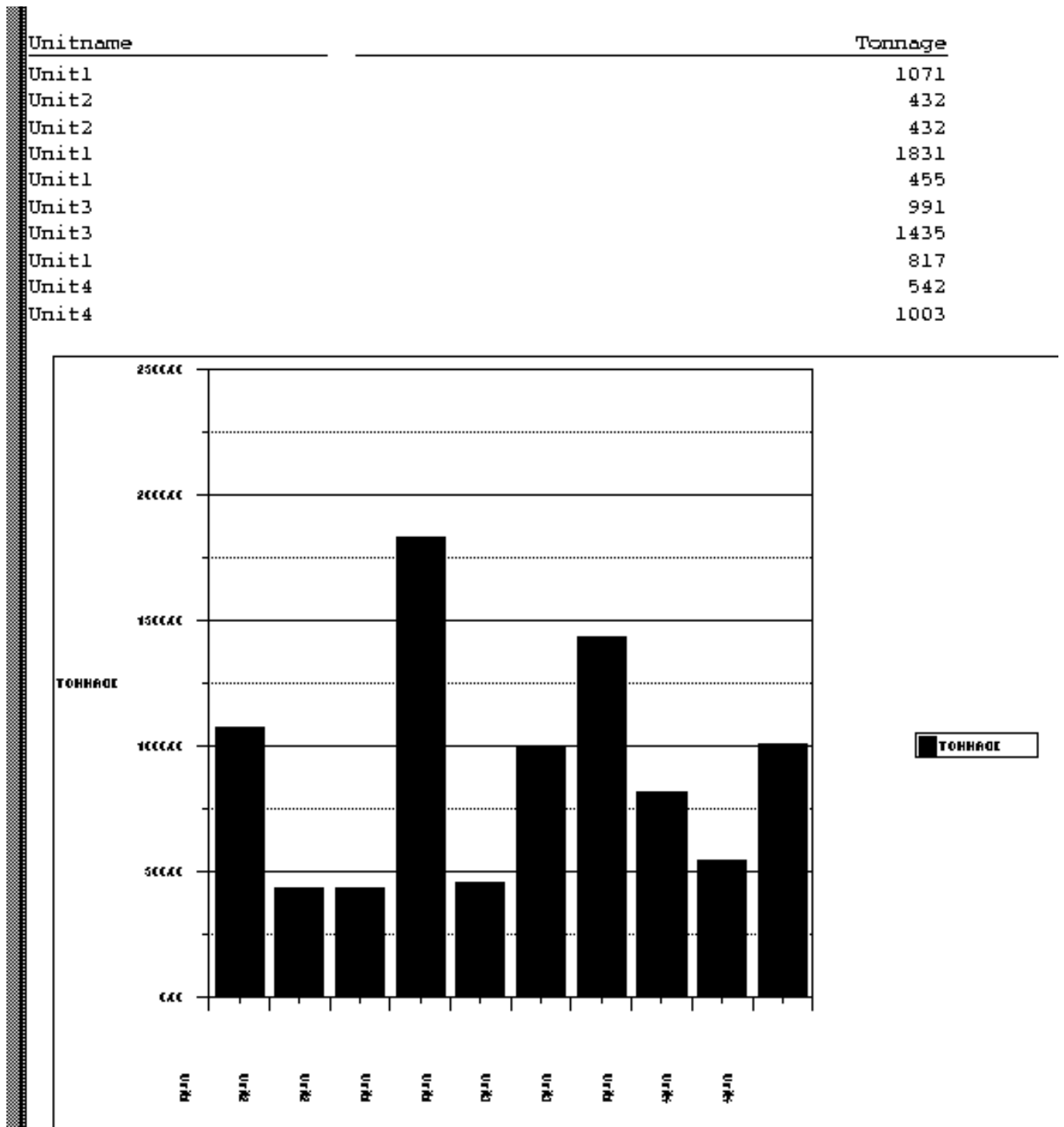


Figure 3-77. Report Preview

3.5.5.2 Tutorial 2 - Integrating Graphics with Queries that Use Functions

In the first tutorial, the query in the report template is a basic select/from statement. When you use more sophisticated queries that perform such functions as summing and grouping, the query you create in the graphic template can not be an exact duplicate of the one in the report template.

This tutorial shows how you can slightly modify the procedure described in [Section 3.5.5.1, Tutorial 1 - Basic Bar Chart](#) to produce a better bar chart where tonnage is summed and grouped by unitname, [Figure 3-78](#).

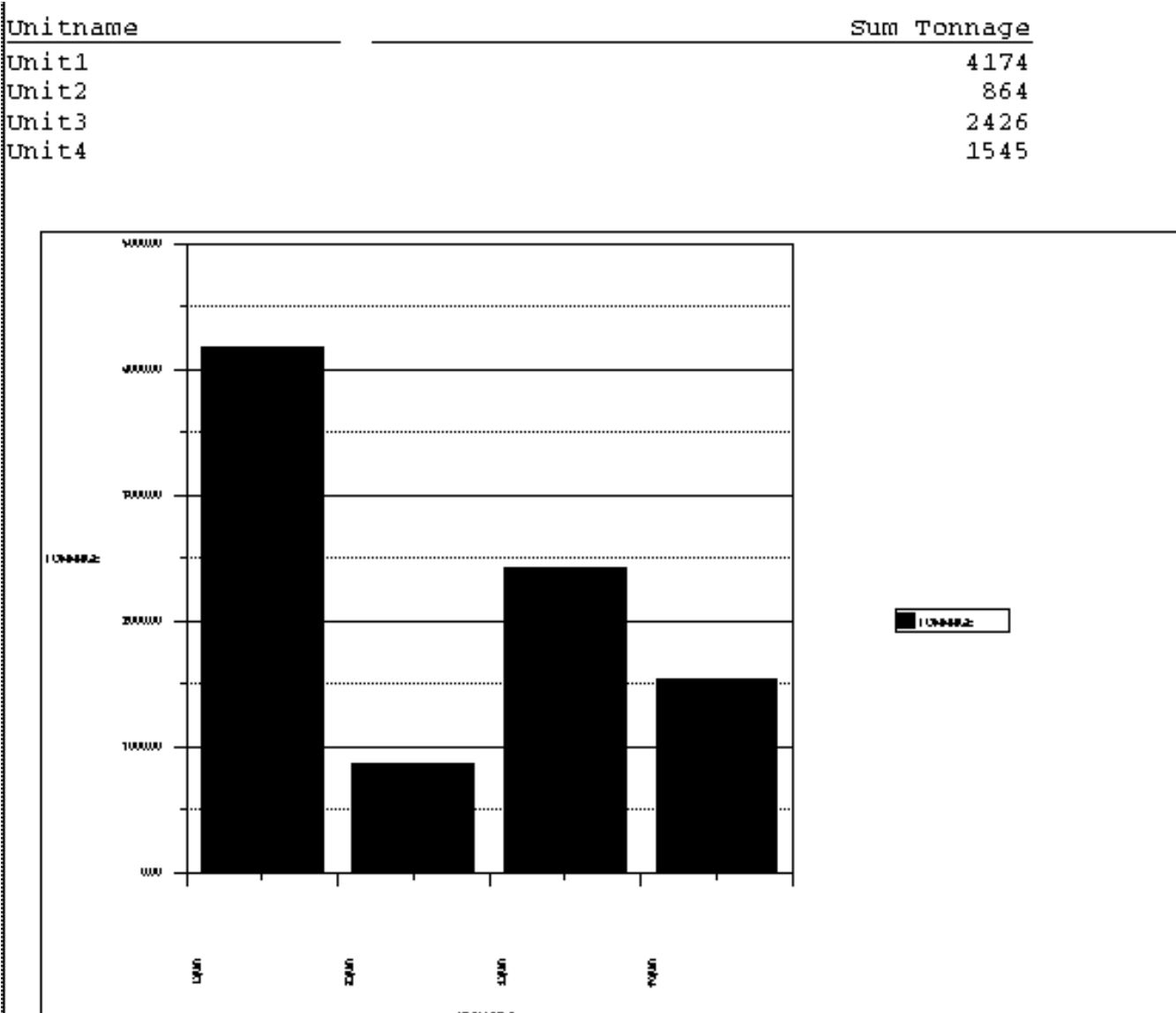


Figure 3-78. Improved Bar Chart

The following is a quick summary of what you need to do different from Tutorial 1:

1. Build a report template with the query shown in [Figure 3-79](#).

The screenshot shows a dialog box titled "Untitled: Query - Q_1". It has a menu bar with "File", "Edit", "Windows", and "Help". Below the menu bar are two tabs: "General" and "Comment". The "General" tab is selected and contains the following fields and buttons:

- Name:** A text box containing "Q_1".
- Maximum Rows:** A text box.
- External Query:** A text box.
- Tables/Columns...:** A button next to the Maximum Rows field.
- Browse...:** A button next to the External Query field.
- SELECT Statement:** A large text area containing the SQL query: `select unitname, sum (tonnage) from example1 group by unitname`.
- Buttons:** "OK", "Apply", and "Close" are located at the bottom of the dialog.

Figure 3-79. Sample Query with Summing and Grouping

2. Build the graphic template, using a simplified version of the same query, [Figure 3-80](#).

NOTE

You can also re-use the Graphic Template that you built in the first tutorial.

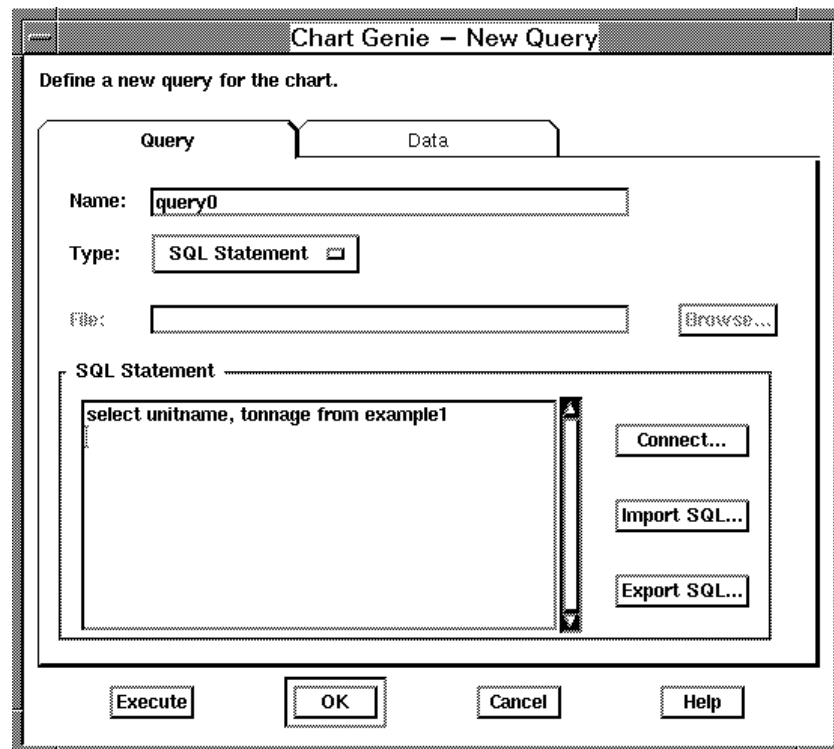


Figure 3-80. Simplified version of Query for Graphic

- When you select the columns on the O.G. Query page of the Oracle Graphics Display dialog, (sum_tonnage and unitname) , for sum_tonnage, click in the Display Column field and enter tonnage, [Figure 3-81](#).

The screenshot shows the 'example: Oracle Graphics Display - D_1' dialog box with the 'O.G. Query' tab selected. The 'Name' field contains 'D_1'. Below the tabs, the 'Display Query' field contains 'query0' and the 'Source' dropdown shows 'G_1'. A table maps 'Report Column' to 'Display Column':

Report Column:	Display Column:
MODE	
ORIENTATION	
PRINTJOB	
sum_tonnage	tonnage
THOUSANDS	
unitname	

At the bottom are 'OK', 'Apply', and 'Close' buttons.

Figure 3-81. Mapping Report Template Query to Graphic Query

- Remember to save the Display in:
`/home/opt/advant/history/customer/Reports/templates/filename
.ogd`

3.5.6 How to Send Report Output to a File

When you schedule a report, you can specify that the report output be sent to a file. This is applicable for all report output modes. To specify that the report output be sent to a file you must enable the File parameter in the Parameter List Configuration dialog box ([Section 3.6.2.1, Parameter List Configuration Dialog](#)).

To enable the File parameter click on the **File** check box. Then click on the **Modify** button to display a dialog box for specifying the file. The full path is required. Use the file naming conventions of your operating system. You can not enter the file path directly in this field. You must enter it via the dialog box.

NOTE

If you intend to send report output to a PDF or HTML file, it is recommended that the file name extension match the file type (for example **.htm** or **.html** for HTML files, or **.pdf** for PDF files). This is because some remote browsers are not able to display the files properly without these extensions. An example of a file specification with a pdf extension is shown in [Figure 3-82](#).

By default, when you send report output to a file, each new report instance overwrites the existing report instance, so there is only one report instance stored in the file. As an option, you can specify that each report instance be stored in a separate file. To store report instances in separate files you must use the Parameter List Configuration dialog to:

- add the parameters **reportprefix** and **reportsuffix** to the parameter list.
- specify an output file in the File field. This is a temporary file that exists just after execution of the report. You do NOT enter **reportprefix** and **reportsuffix** in this field, and the file name you specify here does not determine the file names for the multiple report instances that will be created.

When you configure the report's parameter list this way, subsequent executions of the report are stored in the directory beneath the default report directory with the name specified by the **reportprefix** parameter. For example, if the **reportprefix** parameter is specified as **Area1**, the report instances would be stored in the directory:

`/home/opt/advant/Reports/customer/reports/Area1`

The file name for each instance has the form: `<reportprefix> <date string> <reportsuffix>`

The **reportprefix** and **reportsuffix** are the same for each report instance. The date & time stamp is used to distinguish each report instance. Refer to the following example.

NOTE

Again, if you intend to store report output as PDF or HTML files, it is recommended that the **reportsuffix** end with the appropriate file name extension to match the file type (for example **.htm** or **.html** for HTML files, or **.pdf** for PDF files). This is because some remote browsers are not able to display the files properly without these extensions. An example of a **reportsuffix** with a pdf extension is shown in [Figure 3-82](#).

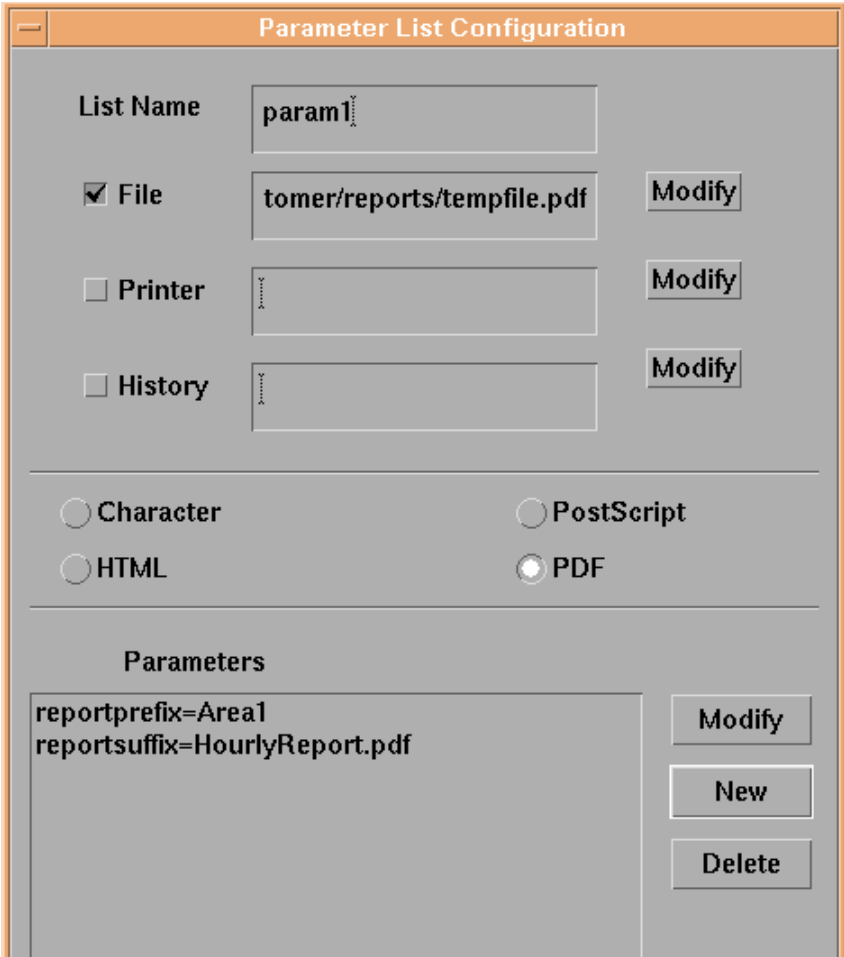
Example

Given the parameter list configuration shown in [Figure 3-82](#) where:

- **File** option is selected and a temporary output file is specified.
- reportprefix = Area1
- reportsuffix = HourlyReport.pdf
- the report is scheduled to execute hourly.

The contents of /home/opt/advant/Reports/customer/reports/Area1 may be:

Area118_Sep_1997_12_00_07HourlyReport.pdf
Area118_Sep_1997_13_00_06HourlyReport.pdf
Area118_Sep_1997_14_00_10HourlyReport.pdf
Area118_Sep_1997_15_00_07HourlyReport.pdf



The image shows a 'Parameter List Configuration' dialog box. It has a title bar with a minus button and the text 'Parameter List Configuration'. Inside, there's a section for 'List Name' with a text field containing 'param1'. Below this are three rows for output options: 'File' (checked), 'Printer' (unchecked), and 'History' (unchecked). Each row has a text field for the output path and a 'Modify' button. The 'File' row's path is '/home/opt/advant/Reports/customer/reports/Area1'. Below these are radio buttons for output format: 'Character', 'PostScript', 'HTML', and 'PDF' (selected). At the bottom is a 'Parameters' section with a text field containing 'reportprefix=Area1' and 'reportsuffix=HourlyReport.pdf', and three buttons: 'Modify', 'New', and 'Delete'.

Figure 3-82. Parameter List Configuration for Multiple Report Instances

The default size (depth) of each reportprefix directory is 50. In other words, the directory will hold up to 50 instances of the report. After 50 instances are stored, each new instance will replace the oldest instance. You can increase or decrease the depth of the directory by including the `reportdepth` parameter in the parameter list, and assigning it an integer value to override the default.

NOTE

Be sure you have enough disk space to store the reports, especially if you increase the report depth or create large reports. Refer to the *Advant Station 500 Series IMS User's Guide* or *Operator Station User's Guide* for information regarding disk space requirements.

Once stored in a file, the report is outside the AdvaInform Reports domain. There are no controls over the modifications that can be done to the file. A report cannot be returned to AdvaInform Reports from the file.

Batch Reports

For batch reports you can use the following set of parameters to store multiple report instances: `batchprefix`, `batchsuffix`, `batchdepth`. These parameters operate in the same manner as the ones described above. The only difference is that the batch reports are distinguished by batch ID rather than a date & time stamp. The file name format will be as follows:

`<batchprefix> <batchid> <batchsuffix>`

where `batchid` is the actual batch ID as seen in the report.

Example

Given:

- the following parameter list configuration:
 - **File** option is selected and a temporary output file is specified.
 - `batchprefix = Unit1`
 - `batchsuffix = .txt`
- `batchid = Batch1, Batch2, Batch3, and so on...`

The contents of `/home/opt/advant/Reports/customer/reports/Unit1` may be as follows:

```
Unit1Batch1.txt
Unit1Batch2.txt
Unit1Batch3.txt
```

NOTE

The ability to send report output to a file is also provided via the Manual Execution window as described in [Section 4.4.4, Manual Execution](#).

3.5.7 How to Print a Report

When you execute a report, you can specify that the report output be printed. This is applicable for all report output modes. Character and PostScript output reports do not require third-party applications to be printed. HTML reports do require a third party application for printing. If you need the ability to launch third-party print routines, you must enter the appropriate command lines in the directory:

```
/home/opt/advant/Reports/user.environment
```

To specify that the report output be printed you must enable the Printer parameter in the Parameter List Configuration dialog box ([Section 3.6.2.1, Parameter List Configuration Dialog](#)). To enable the Printer parameter click on the **Printer** check box. Then click on the **Modify** Button to display a dialog for selecting a printer, or enter the printer name directly if you know the printer name. The printer name is case-sensitive.

3.5.7.1 Printing in the Landscape Mode

When you print a report whose output mode is either Character mode or postscript, you can specify that the report be printed in the landscape mode. To do this, you must include the parameter landscapemode in the report's parameter list. You can assign the parameter any value. This will cause the lp command to have compressed landscape mode options passed., for example: landscapemode = 1.

NOTE

The ability to print a report is also provided via the Manual Execution window as described in [Section 4.4.4, Manual Execution](#).

3.5.8 How to Store Report Output in History

When you execute a report, you can specify that the report output be stored in History as a report log. This is applicable for all report output modes.

To specify that the report output be stored as a report log in History you must enable the History parameter in the Parameter List Configuration dialog box ([Section 3.6.2.1, Parameter List Configuration Dialog](#)). To enable the History parameter click on the **History** check box, then enter the access name for the report log as defined in AdvaInform History. A check will be done that the log really exists.

NOTE

The ability to store reports in History is also provided via the Manual Execution window as described in [Section 4.4.4, Manual Execution](#).

3.6 Configuration/Application Building Menus

This section describes all windows, dialogs, and menus for configuring AdvaInform Reports.

3.6.1 Reports Menu

Access to all AdvaInform Reports functions is provided via the **Reports** menu item in the **AdvaInform** menu in the IMS Menu, [Figure 3-83](#).

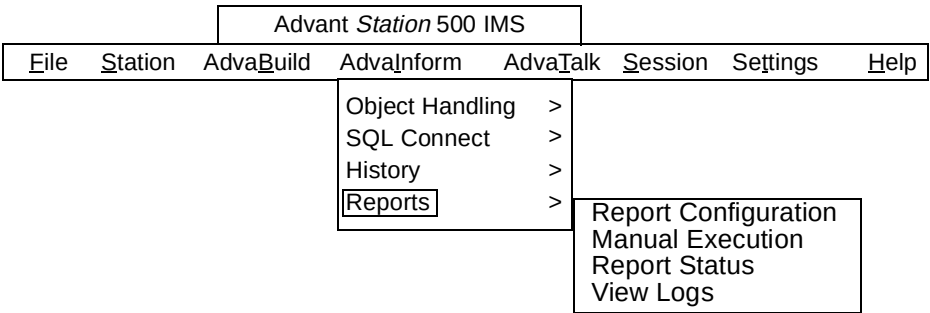


Figure 3-83. AdvaInform Menu Items

This section describes the Report Configuration menu item. The other menu items are related to runtime and are therefore described in Chapter 4 Runtime Operation.

3.6.1.1 Report Configuration

This menu item opens the Report Configuration window. This window provides access to all functions for configuring a report object instance. For further information, refer to [Section 3.6.2, Report Configuration Window](#).

3.6.1.2 Manual Execution

This menu item opens the Manual Execution window. This window lets you execute reports manually, on demand. For further information refer to [Section 4.4.4, Manual Execution](#).

3.6.1.3 Report Status

This menu item opens the Report Status window. This window lists the reports and shows their respective statuses. You can also start and stop cyclic report scheduling via this window. For further information refer to [Section 4.4.1, How to View the Report Status](#).

3.6.1.4 View Logs

This menu item opens the View Logs window. This window provides a listing of report logs. For further information refer to [Section 4.4.5, How to Manipulate Reports in Historical Storage](#).

3.6.2 Report Configuration Window

All functions for configuring AdvaInform Reports are accessed via the Report Configuration window, [Figure 3-84](#).

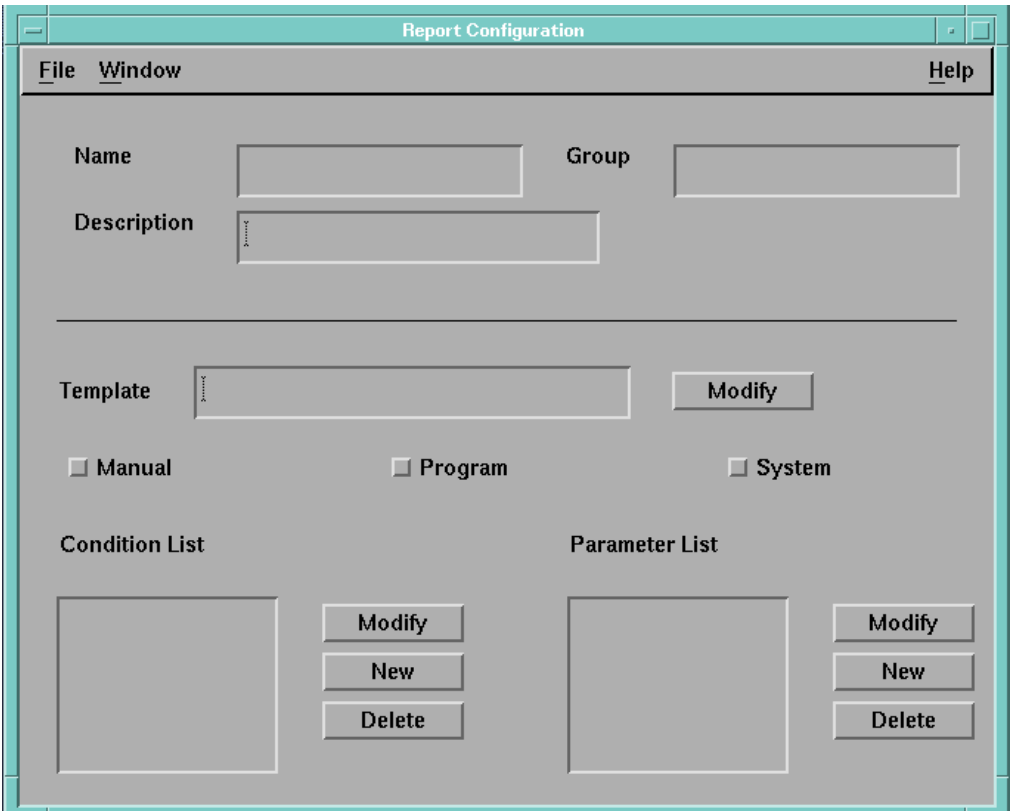


Figure 3-84. Report Configuration Window

To open this window, log in to the IMS Menu as described in [Section 3.3, Application Start-up](#), and choose **Report** from the **AdvaInform** menu. When the **Report** submenu is displayed, choose **Report Configuration**.

The purpose and operation of the fields and buttons in this window are described briefly below. Their operation is described in detail in the sections that follow.

Name Field	This is the name of report the object instance (20 characters max.). Allowed characters are: a - z, A - Z, 0 - 9, dash (-), and underscore (_). You can not edit this field directly. To edit this field, choose New from the File menu to start the New Report dialog. Refer to
Group Field	Report group that the report object instance is assigned to. 20 characters maximum. You can not edit this field directly. To edit this field, choose New from the File menu to start the New Report dialog. Refer to

Description Field	The descriptor is optional. You can enter a descriptor to further identify the report. 28 characters maximum. All characters including spaces are allowed. You can edit this field directly in the Report Configuration window.
Template Field	Assign a report template to the report object instance by entering the report template name in the Template field. Use the name assigned to the template when you built it via Oracle Reports (template name is case-sensitive). The available templates are stored in directory: /home/opt/advant/Reports/customer/templates. Executable templates in this directory have the following file name extension: .rdf If you are not sure of the template name, you can use the corresponding Modify button to display a list of available templates.
Manual, Program, & System Buttons	These check boxes are used to select scheduling options for the report. A shaded button indicates an enabled (selected) option. When Manual is enabled, you can schedule the report object instance via the Manual Execution window. If System is enabled, the report instance is scheduled according to the conditions defined in the Condition List of the Report Configuration window. When Program is enabled, the report can be started by a program. For example, in a MOD 300 application a TCL REPORT statement can start the report.
Parameter List Box	This is where you specify actual values for execution parameters. This includes execution parameters that occur in the assigned report template, and the standard execution parameters for determining the report destination.
New Button for Parameter List	This button displays the Parameter List Configuration dialog to define a new parameter list. Refer to Section 3.6.2.1, Parameter List Configuration Dialog for details.
Modify Button for Parameter List	To modify an existing parameter list, select the parameter list in the Parameter List box, and then click on the Modify button. This displays the Parameter List Configuration dialog box for the selected parameter list.
Delete Button for Parameter List	To delete an existing parameter list, select the parameter list in the Parameter List box, and then click on the Delete button. This displays a message box with a prompt to verify whether or not you want to delete the parameter list. Click OK to verify, or Cancel to cancel the delete operation.
Condition List Box	This is where you define conditions which determine when to execute cyclic reports. The condition list also gives the name of the parameter list to be used for the report execution.

New Button For Condition List	This button displays the Execution Configuration dialog box (for MOD 300) or Execution Type Select box (for Master) to select type of condition.
Modify Button for Condition List	To modify an existing condition list, select the condition list in the Condition List box, and then click on the Modify button. This displays the Execution Configuration dialog box for the selected condition list.
Delete Button For Condition List	To delete an existing condition list, select the condition list in the Condition List box, and then click on the Delete button. This displays a message box with a prompt to verify whether or not you want to delete the condition list. Click OK to verify, or Cancel to cancel the delete operation.

3.6.2.1 Parameter List Configuration Dialog

You use the Parameter List Definition dialog to create a parameter list for a report object instance. The parameter list specifies:

- Report Output Destination - You can specify that the report be output to any one or a combination of the following:
 - File
 - Printer
 - History
- Report Output Mode - You can specify one of four output formats for reports:
 - Character Mode - Use this format when you do not require access via a web browser and the report does not contain graphics.
 - PostScript Mode - Use this format when you do not require access via a web browser and the report includes graphics.
 - HTML or PDF - Use one of these formats when you require access via a web browser. Whether you choose HTML or PDF is a matter of preference, and depends on the software tools that you have available for web browsing.

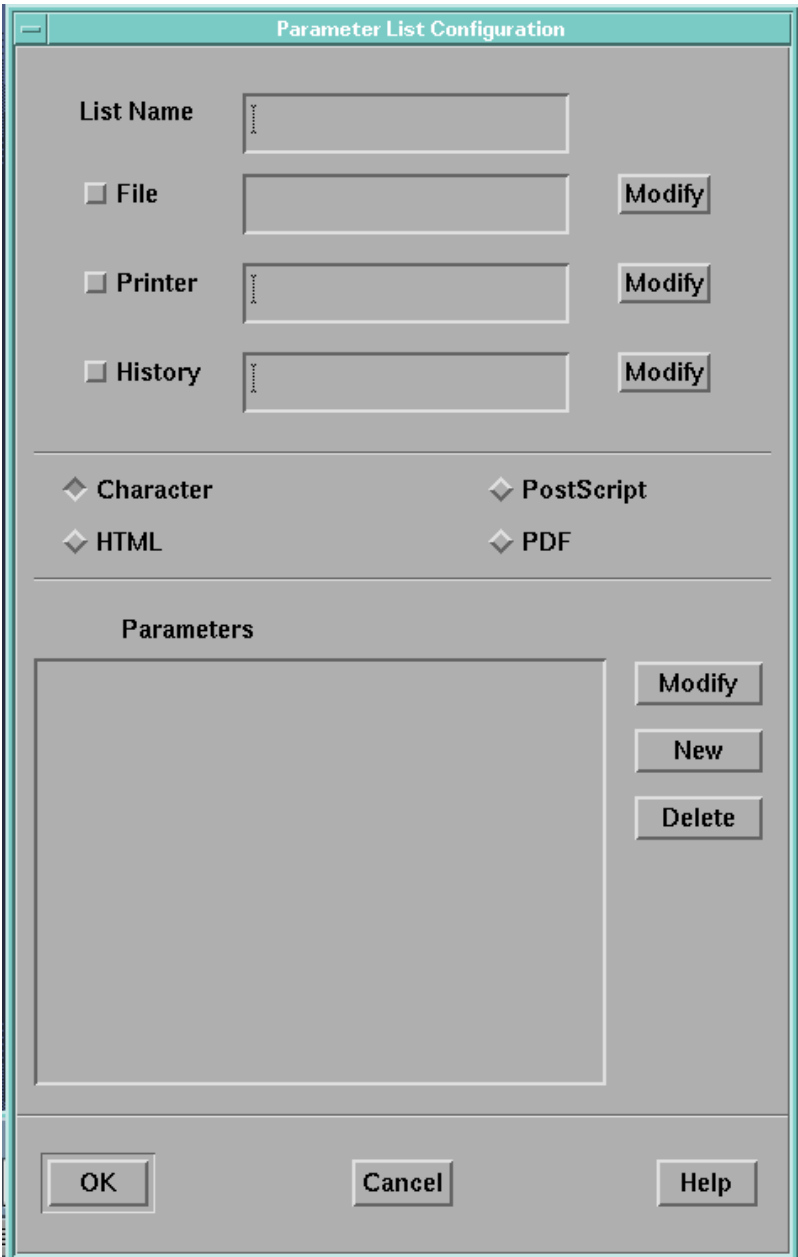
NOTE

Adobe® Acrobat® Reader which supports viewing and printing of PDF files is included with AdvaInform Reports software.

- Execution Parameters - A report template may have parameters that serve as place holders for variables as described in [Section 3.5.2.3, How to Use Execution Parameters in a Query](#). When you create a report object instance, you must specify the values that are substituted for execution parameters when the report executes.

You must define at least one parameter list for each template and you must define at least one parameter to specify the destination for the finished report. You can define more than one parameter list for a template if you want. When you schedule the report, you specify which parameter list to use via the Condition List.

To display the Parameter List Configuration dialog box, [Figure 3-85](#), click on the **New** button by the Parameter List box in the Report Configuration window.



The dialog box is titled "Parameter List Configuration". It contains several sections:

- List Name:** A text input field.
- File:** A checkbox followed by a text input field and a "Modify" button.
- Printer:** A checkbox followed by a text input field and a "Modify" button.
- History:** A checkbox followed by a text input field and a "Modify" button.
- Character:** A diamond-shaped icon followed by the text "Character".
- PostScript:** A diamond-shaped icon followed by the text "PostScript".
- HTML:** A diamond-shaped icon followed by the text "HTML".
- PDF:** A diamond-shaped icon followed by the text "PDF".
- Parameters:** A large empty rectangular box.
- Buttons:** To the right of the "Parameters" box are three buttons: "Modify", "New", and "Delete".
- Footer:** At the bottom are three buttons: "OK", "Cancel", and "Help".

Figure 3-85. Parameter Definition List Dialog Box

This dialog box has the following fields and buttons:

List Name	Name of Parameter List (up to 20 characters maximum)
File	Use this field to specify a file to send the finished report to. The file format can be Character (ASCII), postscript, PDF, or HTML as specified by the chosen report output mode. To enable the File parameter click on the File check box. Then click on the Modify button to display a dialog box for specifying the file, Figure 3-86. The full path is required. Use the file naming conventions of your operating system. You can not enter the file path directly in this field. You must enter it via the dialog box. Once stored in a file , the report outside the AdvaInform Reports domain. There are no controls over the modifications that can be done to the file. A report cannot be returned to AdvaInform Reports from the file.
Printer	Use this field to specify a printer to send the finished report to. To enable the Printer parameter click on the Printer check box. Then click on the Modify Button to display a dialog for selecting a printer, or enter the printer name directly if you know the printer name. The printer name is case-sensitive. Character mode and Graphic mode reports do not require third-party applications to be printed. HTML and PDF reports require third party applications for printing. If you need the ability to launch third-party print routines, you must enter the appropriate command lines in the directory: <code>/home/opt/advant/Reports/user.environment</code>
History	Use this field to specify a report log to send the finished report to. The file format can be Character, postscript, PDF, or HTML as specified by the chosen report output mode. To enable the History parameter click on the History check box, then enter the access name for the report log as defined in AdvaInform History. A check will be done that the log really exists.
Output Mode: Character, PostScript, PDF, or HTML	Use Character mode when you do not require access via a web browser and the report does not contain graphics. If you choose Character mode, you must also select Character mode for report output in Oracle Reports as described in Section 3.3.3, How to Configure the Report Output for Character Mode. Use PostScript when you do not require access via a web browser and the report includes graphics. Use HTML or PDF when you require access via a web browser. Whether you choose HTML or PDF is a matter of preference, and depends on the software tools that you have available for web browsing.

Parameter Box	The Parameter box allows you to specify values for the execution parameters in the report template assigned to the report object instance.
New	This button displays the Parameter Edit dialog to add a new parameter and assign a value to it. Refer to Section 3.6.2.3, Parameter Edit Dialog Box for details.
Modify	To modify an existing parameter, select the parameter in the Parameter List box, and then click on the Modify button. This displays the Parameter Edit dialog box for the selected parameter.
Delete	To delete an existing parameter, select the parameter in the Parameter List box, and then click on the Delete button. This displays a message box with a prompt to verify whether or not you want to delete the parameter. Click OK to verify, or Cancel to cancel the delete operation.

3.6.2.2 File Selection Dialog Box

The File Selection dialog box, [Figure 3-86](#), lets you specify a file to sent a report to.

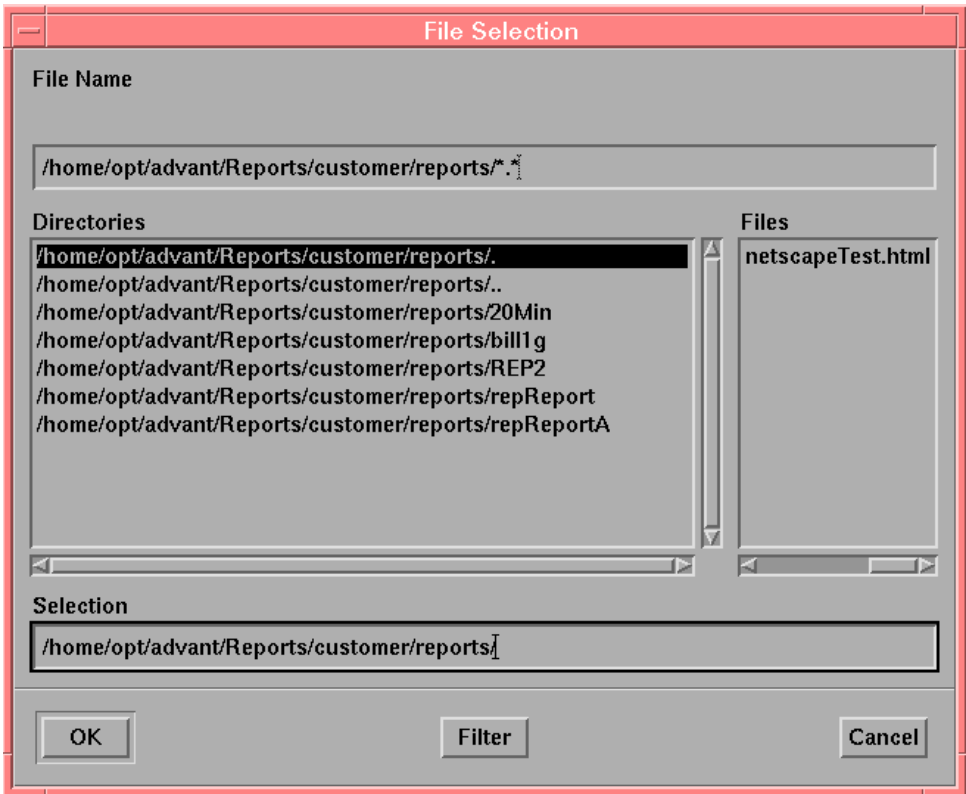


Figure 3-86. File Selection Dialog Box

The fields for this dialog box are described below:

File Name	This is a filter to specify which files will be displayed in the Files selection box. For instance, *.* means all files in the specified directory.
Directories	Use the Directories selection box to specify the path for the files.
File	Use this selection box to select a specific file.
Selection	This box shows the full path specification of the selected file.

It is recommended that you use the default path (/home/opt/advant/Reports/... and so on). [Figure 3-87](#) shows a Unix example.

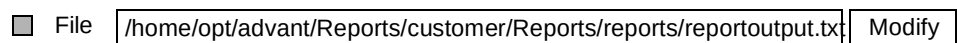


Figure 3-87. UNIX Example of Path and File name

3.6.2.3 Parameter Edit Dialog Box

The Parameter Edit dialog box, [Section 3-88, Parameter Edit Dialog Box](#), is used to add a new parameter to the parameter list, or to modify an existing parameter. To add a new parameter and specify a value for it, click on the **New** Button in the Parameter List Configuration dialog. To modify an existing parameter, first click on the parameter in the list that you want to modify, and then click on the **Modify** button.

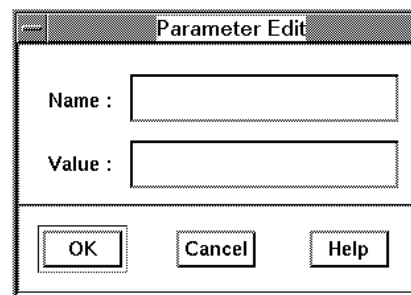


Figure 3-88. Parameter Edit Dialog Box

The operation is basically the same in both cases. If you are adding a new parameter, enter a parameter name in the name field. The Name is the name of the parameter as defined in the report template via Oracle Reports. Enter the value in the Value field. The Value is the value you want to substitute for the parameter when the report executes.

NOTE

If the value includes embedded spaces, the value must be entered in single quotes.
For example: '02/21/96 09:00:00'.

3.6.2.4 Execution Type Select Dialog Box (Master Only)

When you click the **New** button next to the Condition List box, the Execution Type Select dialog box is displayed, [Figure 3-89](#). This dialog box has four buttons for selecting the execution mode (Cyclic or Event),

To start defining a condition, select **New Cyclic** to configure Cyclic execution or **New Event** button to configure Event execution.

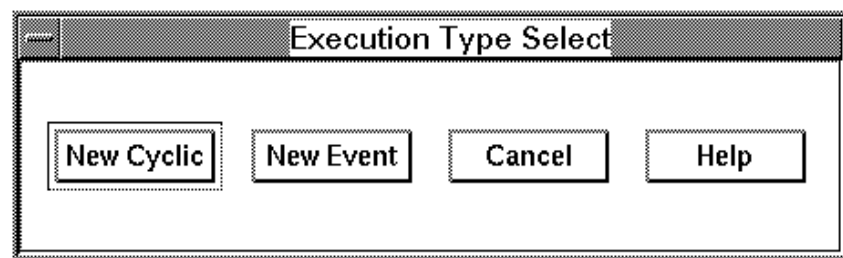
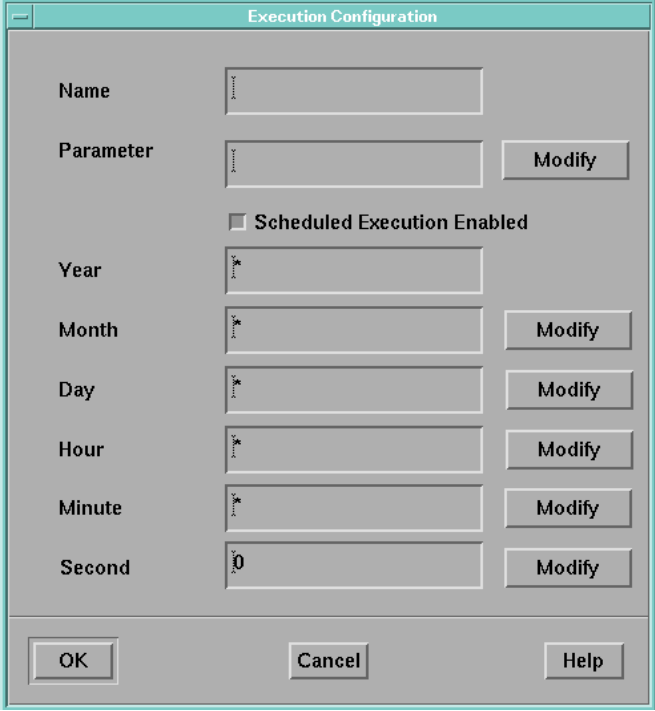


Figure 3-89. The Execution Type Select window

3.6.2.5 Execution Configuration Dialog Box

Scheduling parameters for a report object instance that executes cyclically are specified in a condition list. The condition list also specifies which parameter list to use when you execute the report.

To begin definition of a new condition list, click on the **New** button next to the Condition List. This displays the Execution Configuration dialog box, [Figure 3-90](#).



The dialog box is titled "Execution Configuration". It contains the following fields and controls:

- Name:** A text input field.
- Parameter:** A text input field with a "Modify" button to its right.
- Scheduled Execution Enabled:** A checkbox.
- Year:** A text input field.
- Month:** A text input field with a "Modify" button to its right.
- Day:** A text input field with a "Modify" button to its right.
- Hour:** A text input field with a "Modify" button to its right.
- Minute:** A text input field with a "Modify" button to its right.
- Second:** A text input field with a "Modify" button to its right.

At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

Figure 3-90. Execution Configuration Dialog Box

How to Read the Schedule

When reading a condition:

- An asterisk in a field denotes *every*.
- LAST in the day field denotes last day of the month.
- The day condition is made of two parts in the format:

day of week, day of month

The two parts combine to make the specification. The example below means that the report can execute on Fridays which fall between the first and last days of the month, hence any Friday.

Fri, 1-last

The example below means the report executes on Fridays that fall between the first and the twentieth day of the month.

Fri, 1-20

The example below means any weekday, i.e., the report is enabled from Monday to Friday, where those Mondays to Fridays can be any day of the month.

Mon - Fri, 1 - last

The fields and buttons in this dialog box are described below:

Name	Name of the condition List (up to 20 characters maximum).
Parameter	Name of parameter list to use when you execute the report. Initially, the parameter list is defined as default. However, there is no default parameter list. You must enter the name of a parameter list in this field. Do not leave the parameter list defined as DEFAULT. This has no meaning. If you enter a parameter list name that does not exist for the current report, an asterisk (*) is inserted before the condition list name in the Report Configuration window, Figure 3-91. This indicates the condition list has a parameter list defined that does not exist for the current report instance You can enter the name directly, or click on Modify to display a parameter list selection box. Note that many condition lists using different parameter lists may be executed simultaneously.
Scheduled Execution Enabled	When this check box is selected, the report will execute according to the scheduling parameters. When scheduled execution is disabled, the report will not execute according to the schedule.
Year	The default is all years as indicated by an asterisk. Leave the default if you want the report to execute all years (year-after-year). If you want to select specific years, enter the years directly into the Year field using the following format: 1994 - for one year 1994, 1995, 1996 - for multiple years
Month	The default is to execute the report in all months as indicated by the asterisk. To select specific months, click on the Modify button by the Month field. This displays the Month Configuration dialog box. Refer to Section 3.6.2.6, Month Configuration Dialog Box .
Day	The default is all days as indicated by asterisk. To select specific days, click on the Modify button by the Day Field. This displays the Day Configuration dialog box. Refer to Section 3.6.2.7, Day Configuration Dialog Box .
Hour	Specify the hours of the day. The default is 0 which means 12 O'clock midnight (00:00 for 24-hour clock). To change the default, click on the Modify button by the Hours field. This displays the Hour Configuration dialog box. Refer to Section 3.6.2.8, Hour Configuration Dialog Box .

Minute	Specify the minutes of the hour. The default is 0 (zero minutes past the hour). To change the default, click on the Modify button by the Minute field. This displays the Minutes configuration dialog box. Refer to Section 3.6.2.8, Hour Configuration Dialog Box .
Second	Specify the seconds of the minute. The default is 0 (zero seconds past the minute). To change the default, click on the Modify button by the Second field. This displays the Seconds configuration dialog box. Refer to Section 3.6.2.8, Hour Configuration Dialog Box .

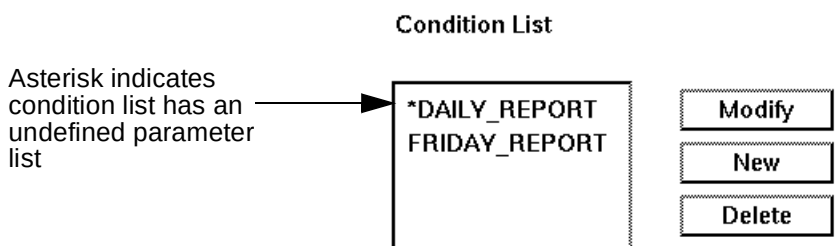


Figure 3-91. Condition List w/Undefined Parameter List

3.6.2.6 Month Configuration Dialog Box

The Month Configuration dialog box is shown in [Figure 3-92](#). Choose a month by selecting the box by its name. More than one month can be specified. You can use the **All** button to select every month. If you want to start over, click on the **None** button. When you are finished with this dialog, click on **OK** to accept the entries.

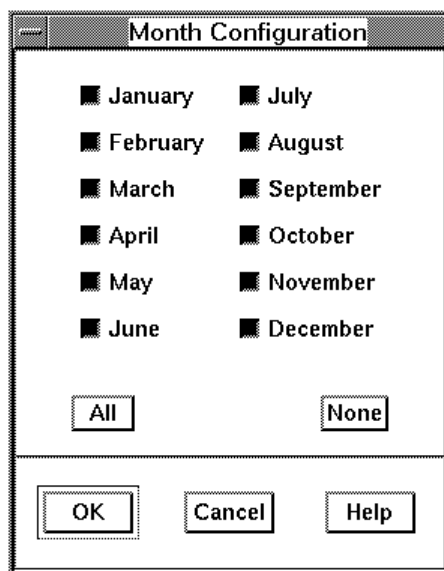


Figure 3-92. Month Configuration DIALOG BOX

3.6.2.7 Day Configuration Dialog Box

The Day Configuration dialog box is shown in [Figure 3-93](#).

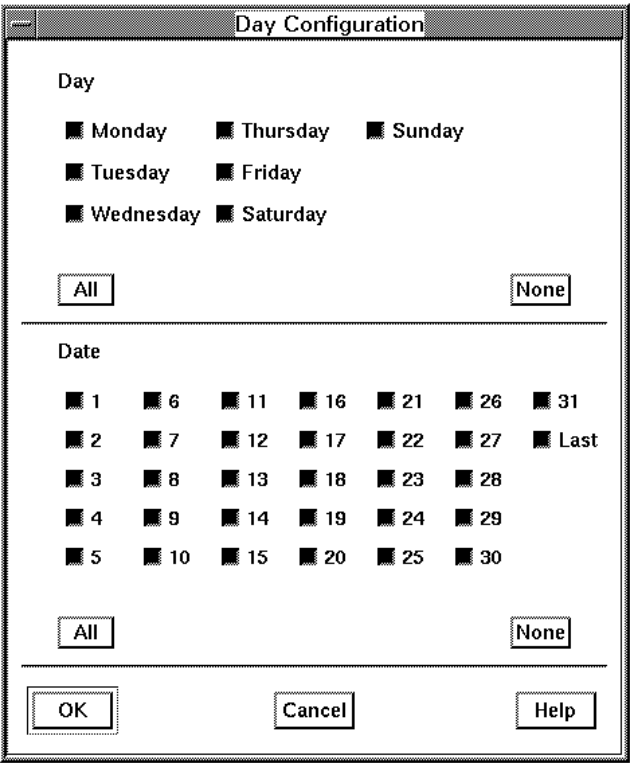


Figure 3-93. Day Configuration Dialog Box

You select the day(s) of the week in the top part of this dialog box, and the day(s) of the month in the bottom part. The default condition for both parts is all days selected. Each part has a **None** button to de-select all days, and an **All** button to select all days.

You use both parts in combination. For instance, you can select **All** in the top part and **1**, **15**, and **Last** in the bottom part to execute the report the 1st, 15th, and last day of the month, [Figure 3-94](#).

The 'Day Configuration' dialog box is divided into two main sections: 'Day' and 'Date'.
The 'Day' section contains checkboxes for the days of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. Below these checkboxes are two buttons: 'All' and 'None'.
The 'Date' section contains a grid of checkboxes for each day of the month (1 through 31) and a 'Last' checkbox. Below this grid are also 'All' and 'None' buttons.
At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Day						
☒ Monday	☒ Thursday	☒ Sunday				
☒ Tuesday	☒ Friday					
☒ Wednesday	☒ Saturday					
All		None				

Date						
☒ 1	☐ 6	☐ 11	☐ 16	☐ 21	☐ 26	☐ 31
☐ 2	☐ 7	☐ 12	☐ 17	☐ 22	☐ 27	☒ Last
☐ 3	☐ 8	☐ 13	☐ 18	☐ 23	☐ 28	
☐ 4	☐ 9	☐ 14	☐ 19	☐ 24	☐ 29	
☐ 5	☐ 10	☒ 15	☐ 20	☐ 25	☐ 30	
All		None				

OK	Cancel	Help
-----------------------------------	---------------------------------------	-------------------------------------

Figure 3-94. Example, 1st, 15th, And Last Day Of Month

When you are finished with this dialog, click on **OK** to accept the entries.

3.6.2.8 Hour Configuration Dialog Box

The Hour Configuration dialog box is shown in [Figure 3-95](#).

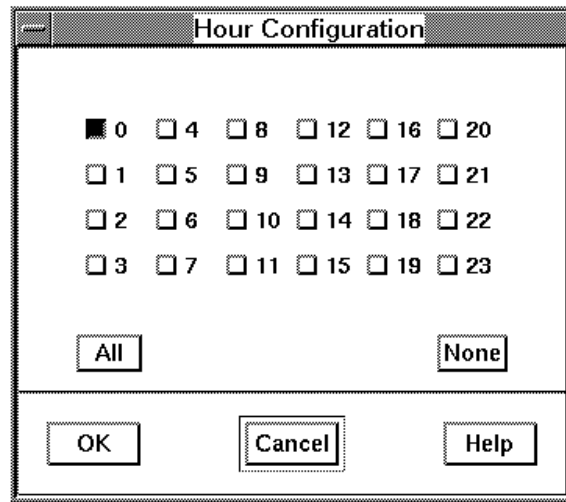


Figure 3-95. Hour Configuration Dialog Box

The default selection is 0 (00:00 for 24-hour clock, or 12 O'clock midnight).

You can select one or more hours as required. Click on the corresponding box to select the hour. If you click on a selected box, that hour is de-selected. You can use **None** to de-select all selected hours, or **All** to select all hours. Click on **OK** when you are finished.

3.6.2.9 Minute Configuration Dialog Box

The Minute Configuration dialog box is shown in [Figure 3-96](#).

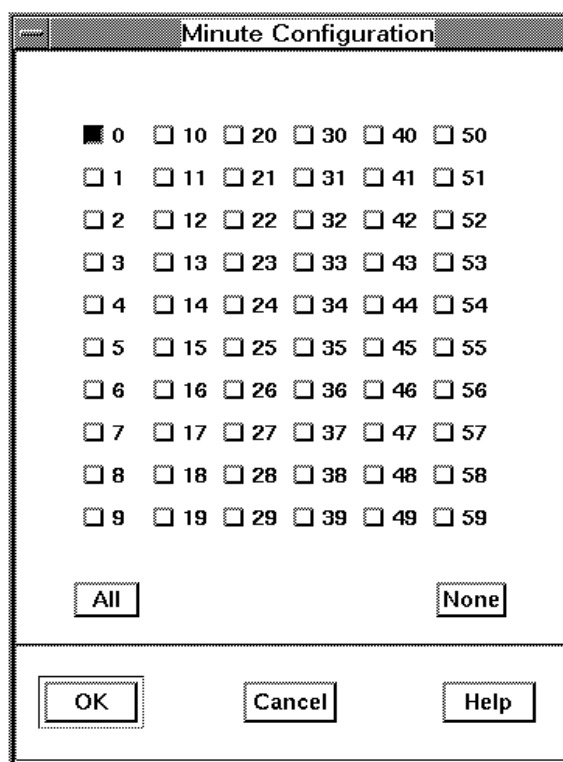


Figure 3-96. Minute Configuration Dialog Box

The default selection is 0. You can select one or more minutes after the hour. Click on the corresponding box to select the minute. If you click on a selected box, that minute is de-selected. You can use **None** to de-select all minutes, or **All** to select all hours. Click on **OK** when you are finished.

3.6.2.10 Second Configuration Dialog Box

The Seconds Configuration dialog box is shown in [Figure 3-96](#).

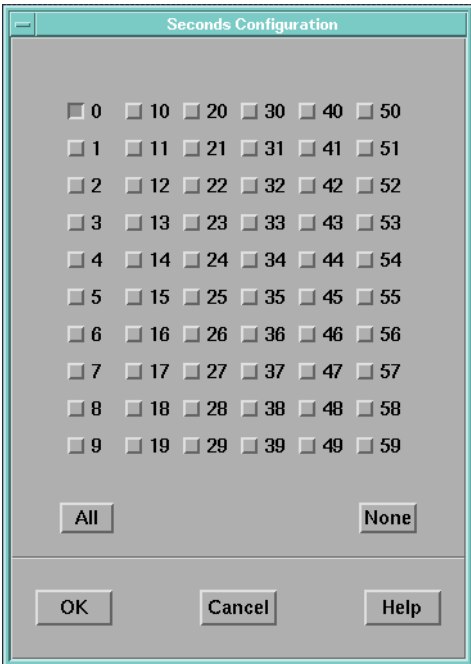


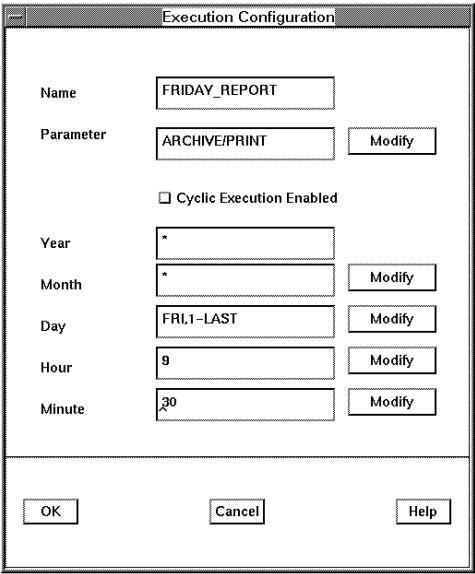
Figure 3-97. Seconds Configuration Dialog Box

The default selection is 0. You can select one or more seconds after the minute. Click on the corresponding box to select the seconds. If you click on a selected box, that second is de-selected. You can use **None** to de-select all seconds, or **All** to select all seconds. Click on **OK** when you are finished.

NOTE

Selecting **All** may have a negative impact on performance and; therefore, is NOT recommended.

Figure 3-98 shows an example of a completed condition.



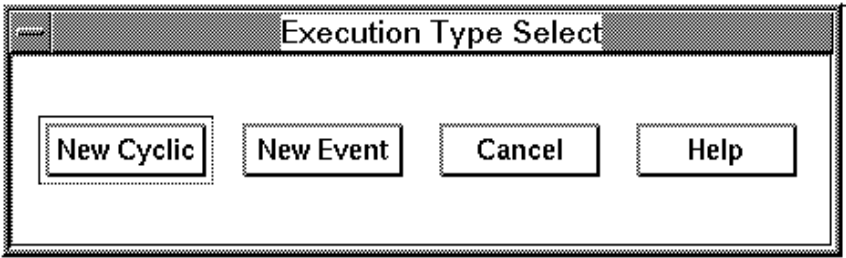
The 'Execution Configuration' dialog box contains the following fields and controls:

- Name:** FRIDAY_REPORT
- Parameter:** ARCHIVE/PRINT, with a **Modify** button.
- Cyclic Execution Enabled:** An unchecked checkbox.
- Year:** * (text field)
- Month:** * (text field), with a **Modify** button.
- Day:** FRI1-LAST (text field), with a **Modify** button.
- Hour:** 9 (text field), with a **Modify** button.
- Minute:** 30 (text field), with a **Modify** button.
- Buttons:** OK, Cancel, and Help at the bottom.

Figure 3-98. Completed Execution (Condition) Configuration

3.6.2.11 Defining Event Conditions (Master Only)

It is possible to define one, and only one, Event Condition for each report instance. To define an Event Condition, first click on the **New** button next to the Condition List in the Report Configuration display. Then click on the **New Event** button in the Execution Type Select dialog box, Figure 3-99.



The 'Execution Type Select' dialog box contains four buttons:

- New Cyclic
- New Event
- Cancel
- Help

Figure 3-99. Execution Type Select box

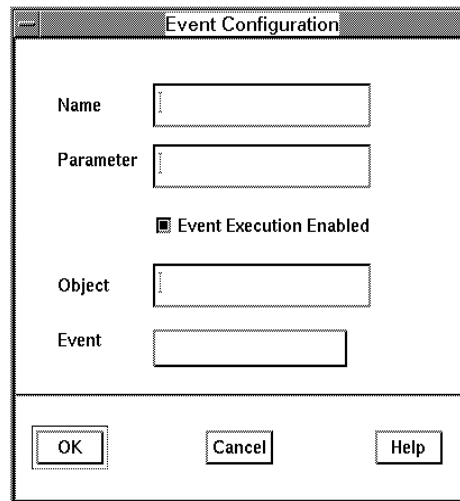
The 'Event Configuration' window is a standard dialog box with a title bar. It contains five text input fields: 'Name', 'Parameter', 'Object', and 'Event'. Between the 'Parameter' and 'Object' fields is a checkbox labeled 'Event Execution Enabled'. At the bottom of the window are three buttons: 'OK', 'Cancel', and 'Help'.

Figure 3-100. Event Configuration window

1. Enter the name of the Event Condition in the Name field
2. Enter the name on the Parameter List that shall be used when the report instances is executed by the event in the Parameter field.
3. Select the Event Execution Enable toggle button to activate the Event Condition for the report instance.
4. Enter the name of the DI object that shall trigger this report instance to execute in the Object field. The DI object name must be resolved before you can use it in an Event condition.
5. Select the Event push key to display the Event list, [Figure 3-101](#).

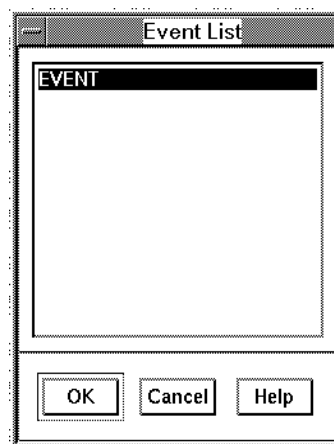
The 'Event List' window is a dialog box with a title bar. It features a list box containing a single item labeled 'EVENT'. At the bottom of the window are three buttons: 'OK', 'Cancel', and 'Help'.

Figure 3-101. Event List

6. Select the event that shall trigger the report instance and then click **OK**.

3.6.2.12 Report Configuration Window - File Menu

File - New

New displays the New Report dialog box, [Figure 3-102](#). This is where you define the Group and Name fields. When you choose **New** all dialog information in the Report Configuration window is cleared. If you have made any changes to the previous instance, a dialog box is displayed on the screen asking if you want to save the changes.

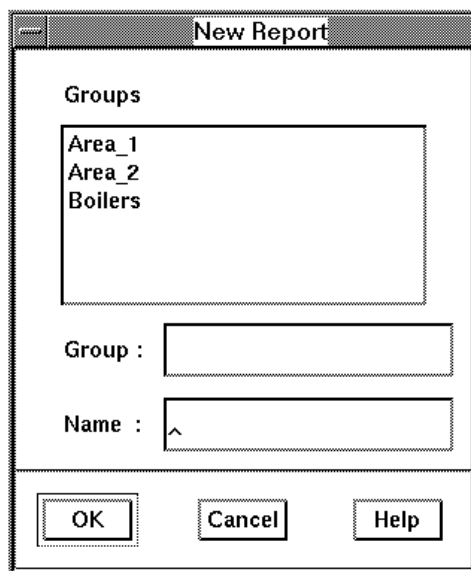


Figure 3-102. Entering the Report Object Instance Group and Name

Any existing groups are displayed in the Groups selection box. If the report object instance is to be assigned to an existing group, you can select that group from the Groups selection box; otherwise, enter a new group name directly into the Group field.

The name you enter in the Name field is the name used to save and retrieve the report instance. Guidelines for naming report instances are:

- Allowed characters are: a - z, A - Z, 0 - 9, dash (-), and underscore (_). Do not use periods.
- Name and Group can each be up to 20 characters maximum.
- Name and Group are case-sensitive. For instance, if you define Name as Boiler_Temps, you can not use boiler_temps.

When you close the New Report dialog box, the group and report instance names are entered in their respective fields in the Report Configuration window.

File - Open

Open displays a listing of report instances in a particular node. From this list you can select the instance you want to open and modify. Because you can have only one report instance opened at a time, you are warned by a dialog box to save any changes to the previous instance.

File - Save

When the report object instance configuration is finished, choose **Save** from the **File** menu in the Report Configuration window. The report instance can now execute. If Manual is enabled, the report can be scheduled from the Manual Execution window. If System is enabled, the report is executed each time one of the specified conditions occur. If Program is enabled, the report can be scheduled by a program such as one developed via the User API or by a REPORT statement in a TCL program.

Save stores the current instance on disk. Choosing **Save** overwrites all old information about the instance. A first time save displays a dialog box for naming the report instance.

File - Save As

Save As displays a dialog box for entering a new name for the instance. If you enter a name that already exists, a message box is displayed which asks you whether or not you want to overwrite the present instance with this name.

Delete

Delete eliminates the currently displayed instance. A dialog box asks for confirmation before deleting.

Exit

Exit closes the Report Configuration window. A message box is displayed which asks if you want to save changes to the current window before exiting.

3.6.2.13 Report Configuration Window - Window Menu

Build Template

Build Templates opens Oracle Reports for building templates.

Build Graphics

Build Graphics opens Oracle Reports for building graphics.

View Logs

View Logs opens the View Logs window.

Manual Execution

Manual Execution opens the Manual Execution window.

Report Status

Report Status opens the Report Status window.

3.7 How to Write Queries for Reports

This section provides guidelines for writing SQL queries for AdvaInform Reports, as well as examples of SQL queries for different applications. You can use the example queries in this section as the basis for queries in your own reports.

3.7.1 Relational Database Structure

SQL queries are written against relational tables or views. Views are custom representations of Oracle tables that let you customize the way you read data from tables. You can create a view by selecting columns from one or more tables to include in the view. Once the view is created, you can read data from the view rather than the tables.

In AdvaInform Reports, the relational database concepts of tables (or views), rows and columns are equated to Advant OCS concepts of Object Type, Object Instance, and Object Attribute as indicated in [Table 3-1](#).

Table 3-1. Relational Database Concepts vs. Object Concepts

Relational Database	Object (ODB)	Comment
Table or view	Object Type	Views of standard Advant OCS object types have been created for you to simplify accessing data from the Advant OCS object types. These views are described in Section 3.7.3, Views of Advant OCS Object Types .
Row	Object Instance	
Column	Attribute	

3.7.2 SQL Query Structure

SQL queries for AdvaInform Reports are based on the SELECT statement and its clauses: INTO, FROM, WHERE, GROUP BY, HAVING, and ORDER BY. The following is a description of the purpose of each clause.

An Example SQL Query

An example of an SQL query for a MOD 300 application is shown in [Figure 3-103](#).

```
SELECT tag, measure, setpoint, engunits
FROM ccf_pid_loop
WHERE name='TIC172'
AND measure > 500
ORDER BY measure
```

Figure 3-103. Example SQL Query for MOD 300

An example of an SQL query for a Master application is shown in [Figure 3-104](#).

```
SELECT name, value  
FROM PIDCON  
WHERE value > 500  
ORDER BY value
```

*Figure 3-104. Example SQL*Connect Query for Master*

NOTE

SQL queries are not case-sensitive, except for text in single quotes. You can write SQL keywords (for example select) in uppercase or lowercase. The same is true for view and column names. Object (loop) names must be entered exactly as they were configured.

The SELECT Clause

The SELECT clause specifies which object attribute(s) to retrieve. An object attribute equates to a column in a row in a relational table or view. Object attributes can be manipulated mathematically. For instance, this retrieves a value equal to the measured value plus 10:

```
SELECT measure + 10
```

The FROM Clause

The FROM clause specifies the name of an Advant OCS object type. The Object Type equates to a relational table or view. Object type names are provided in the *Object Types Reference Manual* or list types within AdvaInform SQL*Connect - Client configuration.

The WHERE Clause

The WHERE clause can be used to specify conditions for the query. This can be used to specify a unique object instance or range of object instances to query. In [Figure 3-103](#) and [Figure 3-104](#), the condition specifies that only objects with a measured value greater than 500 will be retrieved.

The ORDER BY Clause

The ORDER BY clause determines the order that instances are returned in. If you do not specify an order, the order is determined by an internal criterion. If you specify ORDER BY attribute, instances are returned in ascending order. You can specify that instances be returned in descending order by entering DESC after the attribute. For instance:

```
ORDER BY measure DESC
```

3.7.3 Views of Advant OCS Object Types

Table 3-2 describes the views of Advant OCS object types that you can write queries against in AdvaInform Reports. The *AdvaInform Object Types Reference Manual* contains complete listings of these views.

Table 3-2. Pre-defined Tables and Views for Advant OCS Objects

System	Table Name	Purpose
MOD 300	CCF_CONTIN_LOOP	Loop values from continuous loops which have no Auto/Manual or PID Controller FCMs
	CCF_CNTRL_LOOP	Loop and Auto Manual Controller FCM attributes from continuous loops which have Auto/Manual FCMs.
	CCF_PID_LOOP	Loop and PID FCM attributes from continuous loops which have PID FCMs but do not use adaptive reset or adaptive gain.
	CCF_1ADAPT_LOOP	Loop and PID FCM attributes from continuous loops which have PID FCMs and use either adaptive reset or adaptive gain.
	CCF_2ADAPT_LOOP	Loop and PID FCM attributes from continuous loops which have PID FCMs and use both adaptive reset and adaptive gain
	FCM object types	Each FCM type has an object type associated with it. For example, math_fcm, ainp_fcm, and so on.
	CCF_DEVICE_LOOP	Device loop attributes
	DATA_FCM, VA_STRING	Reading values from AdvaInform Basic Objects from either the local or remote nodes.
Master	AI, AO, DI, DO, TEXT, DAT	Reading values from AdvaInform Basic Objects from either the local or remote nodes. AI is used to read AI objects, AO for AO objects, and so on.
Common (MOD 300 & Master)	NUMLOGVAL	Historical values from numeric historical logs
	MODMSGLOG (MOD 300) EAMSGLOG (Master)	For MOD 300, MODMSGLOG contains messages from all historical message logs. For Master, to get a more concise view of EAMSGLOG, use BCHS_MESSAGES.
	ENCAP_DATA	Used by AdvaInform Report users to put data from an executed report into a new report
	IMS_TABLES	Contains names of tables to which the current AdvaInform user has privileges.

3.7.4 How to See Which Attributes are Available in an Object Type

The `describe` function lets you see which attributes are included in a particular object type. This function returns the attribute names and corresponding data type, [Figure 3-105](#).

```
SQL> describe numlogval
```

Name	Null?	Type
NAME		VARCHAR2(82)
TIME	NOT NULL	DATE
MS	NOT NULL	NUMBER(10)
ENTRYVALUE	NOT NULL	FLOAT(24)
ENTRYSTATUS	NOT NULL	NUMBER(10)
OBJECTSTATUS	NOT NULL	NUMBER(10)
RET_TYPE		VARCHAR2(14)
INTERP_ALG		VARCHAR2(6)
SELECTION		VARCHAR2(10)
CALC_ALG		VARCHAR2(14)
INTERVAL		NUMBER(10)

Figure 3-105. Example, Describe Function

You execute the `describe` function from the Database Access window as shown below:

```
describe name
```

where *name* is the name of the object type

For example:

```
describe numlogval
describe ccf_contin_loop (MOD 300)
describe math_fcm (MOD 300)
```

To open the Database Access window, choose **Database Access** from the **Station** menu on the IMS main menu.

3.7.5 Queries for Process Data (MOD 300 Examples)

In the following queries SQL keywords are shown in CAPS for emphasis, however, they can also be written in lowercase.

TIP

When you write queries for CCF loops, the WHERE clause can either use the name attribute or the tag attribute to specify the tag. For example, these queries retrieve the same data:

```
SELECT tag, measure FROM ccf_pid_loop WHERE tag = 'LC200'
SELECT name, measure FROM ccf_pid_loop WHERE name = 'LC200'
SELECT tag, measure FROM ccf_pid_loop WHERE name = 'LC200'
```

It is recommended that you use the name attribute in the WHERE clause for the following reason. The location table identifies object instances by the name attribute. Therefore, the following query can be used to put an entry for LC200 in the location table. In contrast, a query that identifies a loop by its tag attribute does not update the location table.

```
SELECT name, measure FROM ccf_pid_loop WHERE name = 'LC200'
```

3.7.5.1 Continuous Loop with no Controller FCMs

Use the CCF_CONTIN_LOOP view for the basic continuous CCF loops that have no controller FCMs. There are no FCM attributes in this object type.

The following query uses the CCF_CONTIN_LOOP view to get the current measured value for loop calc1.

Query:

```
SELECT measure FROM ccf_contin_loop  
WHERE name = 'CALC1'
```

3.7.5.2 FCM Attributes in Continuous Loop

To read the value of an fcm attribute (for example RESULT) from a continuous loop, use the appropriate fcm view. Each FCM type has an associated FCM view. For example, if you want to read the result of the Analog Output FCM of loop TC50, a possible query is:

The following query uses the aout_fcm view to get the result for TC100.AOT.

Query:

```
SELECT name, result FROM aout_fcm  
WHERE name = 'tc100.aot'
```

TIP

Note that the name attribute for an FCM is in the standard CCF format: tag.fcmname.

3.7.5.3 Auto/Manual Controller Loop

To read information for a loop that has an Auto/Manual FCM use the CCF_CNTRL_LOOP view. This view contains both loop-level attributes and auto/manual controller FCM attributes. However, attributes of the other FCMs of the loop are not in this object type. For example, if you want to read a value from the Analog Input FCM of the loop, you read it from the AINP_FCM view.

The following query uses the CCF_CNTRL_LOOP view to get the current loop state and ratio value for AMC1.

Query:

```
select lp_state, r_value from ccf_cntrl_loop  
where name = 'AMC1'
```

3.7.5.4 PID Controller Loop

Use one of the following views to read values for PID Controller loops, depending on the kind of loop:

CCF_PID_LOOP	Use CCF_PID_LOOP when the loop DOES NOT have any adaptive functions.
CCF_1ADAPT_LOOP	If the loop uses one kind of adaptive function (either adaptive gain or adaptive reset), use CCF_1ADAPT_LOOP.
CCF_2ADAPT_LOOP	If the loop uses both adaptive gain and reset, use CCF_2ADAPT_LOOP.

The following query uses the CCF_PID_LOOP view to get the setpoint for TC500.

Query:

```
SELECT setpoint FROM ccf_pid_loop
WHERE name = 'TC500'
```

TIPS

The PID Controller loop object types contain both loop-level attributes and PID controller attributes. However, attributes of the other FCMs that may be included in the loops are not in these object types. For example, if you want to read a value from the Analog Input FCM of the loop, you read it from the AINP_FCM object type.

When you configure the location table, it is best to associate a PID loop with the proper object type. However, if you associate a loop with an incorrect type, you may be able to read a subset of the loop's data. For example if you associate a PID controller loop with object type CCF_CONTIN_LOOP, you can read the loop level attributes such as Measure and Alarm State, but you cannot read the PID attributes such as Base Gain.

3.7.5.5 Device Loop

Use the CCF_DEVICE_LOOP view to read device loop attributes.

Query:

```
SELECT dev_cmndtxt dev_stattxt FROM ccf_device_loop
WHERE name = 'MIXER_1'
```

TIP

The dev_cmndtxt and dev_stattxt, and field_stattxt attributes in this view can be used to retrieve device command, device state, and field state data directly in text format rather than having to convert it.

3.7.6 Queries for Numeric History Data

Numeric history data is stored in objects called logs. A log stores multiple instances of one object attribute value. For instance, \$HSFC100,MEASURE-1-o stores multiple instances of the measured value for FC100. Each instance is referred to as a log entry, or simply entry. Each entry in a log has a value and a corresponding time stamp as shown below:

Timestamp	Value
25 Jul 95 09:30:00	65.7
25 Jul 95 09:30:10	66.4
25 Jul 95 09:30:20	65.9
25 Jul 95 09:30:30	66.7
and so on...	

To access data from a numeric log via AdvaInform Reports, you write a query against the NUMLOGVAL table which has the attributes shown in [Table 3-3](#).

Table 3-3. NUMLOGVAL Attributes

Attribute	Data Type	Description
NAME	VARCHAR2(82)	Log name or Access Name
TIME	DATE	Date of the stored value
MS	NUMBER(10)	Millisecond time of the stored value
ENTRYVALUE	FLOAT(24)	The stored value
ENTRYSTATUS	NUMBER(10)	The status of the stored value. See <i>AdvaInform History Users Guide</i> .
OBJECTSTATUS	NUMBER(10)	The status of the object that provided the stored value.
RET_TYPE	VARCHAR2 (14)	Retrieval type. Possible values are: RAW - Actual sampled value INTERPOLATE - Interpolated value between samples
INTERP_ALG	VARCHAR2 (6)	Interpolation algorithm. Possible values are: AVE - Average MAX - Maximum MIN - Minimum INST - Instantaneous
SELECTION	VARCHAR2 (10)	When you use the Modify Numeric Value function (USER API), SELECTION can be used to specify current data or original data: CURRENT - Current (modified) value ORIGINAL - Value before modification
CALC_ALG	VARCHAR2 (14)	Used to specify the log whose calculation algorithm matches this specification: AVERAGE MAXIMUM MINIMUM INSTANTANEOUS SUM SUM_OF_SQRS STANDARD_DEV NUM_OF_VALUES WILDCARD
INTERVAL	NUMBER (10)	Interval for data retrieval in seconds. For instance to retrieve values at 1-hour intervals, enter 3600.

3.7.6.1 When to Use Log Name or Access Name

There are two ways to specify which log to retrieve data from. The access name is the name of the data source that you assign to a log when you configure it in the History Builder. For instance, TC100,MEASURE (MOD 300) or TC100,VALUE (Master).

By virtue of the hierarchical log structure of AdvaInform History logs, there can be more than one log using the same data source. To distinguish multiple logs with the same data source, History automatically assigns a unique default log name to each log. The log name is derived from the access name, and has a prefix and suffix appended as shown in [Figure 3-106](#).

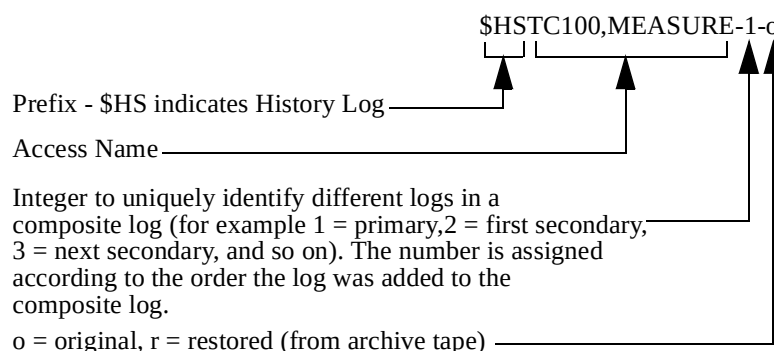


Figure 3-106. Log Name

You can use the access name to specify which log to retrieve data from when you know the data source but do not know the log name. When you use the access name, History uses the seamless retrieval criteria, [Section 3.7.6.8, Criteria for Seamless Retrieval](#), to select the most appropriate log in the data source's log hierarchy. Generally, unless other criteria are specified, seamless retrieval selects the log that has INSTANTANEOUS or AVERAGE as the calculation algorithm, and has a log period that provides the best coverage of the requested time period.

Some example queries for numeric data are provided in the following sections.

3.7.6.2 Basic Query for Numeric Data

This query does not specify a time range or instance. Therefore, it returns all entry values with their respective timestamps for TC100,MEASURE.

Query:

```
SELECT to_char(time, 'mm/dd/yy hh24:mi:ss'), entryvalue
FROM numlogval
WHERE name = 'TC100,MEASURE'
```

TIPS:

The time attribute stores the timestamp for an entryvalue. It is a *date* data type. If you query for time directly (without using `to_char`), you get just the day, month, and year. To get the full timestamp including hours, minutes, and seconds, use the `to_char` function to convert the time attribute from the date data type to the character data type. When you convert time to a character data type, you must specify the format mask. The example above shows one possibility. Some other valid format masks are given below are:

```
SELECT to_char(time, 'dd/mm/yy hh24:mi:ss')
SELECT to_char(time, 'dd-mon-yy hh24:mi:ss')
```

Other format masks can be used. Note that the symbol for minutes is `mi` rather than `mm`.

3.7.6.3 Query for Numeric Data from a Secondary Log

When you query for a secondary log, it is recommended that you use the full log name when you know it. This way you are sure to access the intended log. If you use just the access name, the seamless retrieval function will select the log based on the seamless retrieval criteria as described in [Section 3.7.6.8, Criteria for Seamless Retrieval](#).

This query is identical to the one before, except that it uses the full log name to specify a secondary log.

Query:

```
SELECT to_char(time, 'mm/dd/yy hh24:mi:ss'), entryvalue
FROM numlogval
WHERE name = '$HSTC100,MEASURE-3-o'
```

3.7.6.4 Query for a Specific Time Range

This query is very similar to the basic numeric query. The **WHERE** clause is expanded with two **AND** phrases that specify the time range as: after 13:34 (1:34 pm) on February 15, 1996, and before 13:37 (1:37 pm) on February 15, 1996.

Query:

```
SELECT to_char(time, 'mm/dd/yy hh24:mi:ss'), entryvalue
FROM numlogval
WHERE name = 'TC100,MEASURE'
AND time>=to_date('02/15/96 13:34:00', 'mm/dd/yy hh24:mi:ss')
AND time<=to_date('02/15/96 13:37:00', 'mm/dd/yy hh24:mi:ss')
```

TIP:

The time range is specified as a character data type, and must be converted to a date data type in order to be compared to **time**. This conversion is performed by the **to_date** function. The format mask must be specified (see [Section 3.7.6.2, Basic Query for Numeric Data](#)).

3.7.6.5 Query for Interpolated Data

An interpolation feature of the numlogval view lets you query for data at fixed intervals such as every five, ten, or fifteen minutes, even if data is not sampled and stored at such intervals.

The following example shows how to read interpolated values from a numeric log with a time interval between the values of two hours and a time span of 12 hours. Note that you have to specify **INTERVAL**, **TIME < xxx**, and **TIME > yyyy** in the where clause.

Query

```
SELECT TO_CHAR (TIME, 'DD/MM/YYYY HH:MI:SS'), ENTRYVALUE FROM
NUMLOGVAL

WHERE NAME = '$HSTC100,VALUE-1-o' AND

RET_TYPE = 'INTERPOLATED' AND

INTERP_ALG = 'AVG' AND

INTERVAL = 2*60*60 AND

SELECTION = 'CURRENT' AND

TIME > TO_DATE('19960311 8:00:00', 'YYYYMMDD HH24:MI:SS') AND

TIME < TO_DATE('19960311 20:00:00', 'YYYYMMDD HH24:MI:SS');
```

3.7.6.6 Query Based on Calculation Algorithm

If you want to retrieve logs that are created with a specific calculation algorithm you can specify `calc_alg` together with the access name in the select statement. The `calc_alg` column can only be used in conjunction with the access name.

```
SELECT TO_CHAR (TIME, 'DD/MM/YYYY HH:MI:SS'), ENTRYVALUE FROM  
NUMLOGVAL  
  
WHERE NAME ='TC100,VALUE' AND CALC_ALG = 'MAX';
```

3.7.6.7 Query for a Specific Instance in Time

Again, this query is very similar to the ones before. In this case, the query will return entry values from eight hours ago to the present.

Query

```
SELECT to_char(time, 'mm/dd/yy hh24:mi:ss'), entryvalue  
FROM numlogval  
WHERE name = '$HSTC100,MEASURE-1-o'  
AND time> (sysdate -8/24)
```

TIP

The `sysdate` function provides the system date and time. It is the date datatype. You can subtract days or hours from the system time as follows:

- to subtract hours, use the format `n/24` and then subtract the fraction from `sysdate` as in the example above.
- to subtract days, subtract an integer value from `sysdate`. For example:

```
AND time > (sysdate - 8)
```

This query returns entry values from eight days ago to the present.

3.7.6.8 Criteria for Seamless Retrieval

The seamless retrieval function uses the following criteria to search the History database and select a log when a request for history data is issued:

1. Sort logs by percentage of the time period of the request that the log covers, and choose the set of logs that are closest in coverage.

For instance, if the request is for 24 hours, logs may be sorted as follows in descending order of preference: 24-hour logs and 168-hour logs, 8-hour logs, and 1-hour logs. Since the 24-hour logs and 168-hour logs cover 100% of the requested time period, they would be chosen.

2. Search the set chosen in step 1, and find all logs that use the specified calculation algorithm. Logs that do not use the specified algorithm are dropped from consideration.

For AdvaInform Reports, the algorithm is always WILDCARD which uses the following order of preference:

- a. AVERAGE or INSTANTANEOUS

These algorithms have equal preference. The first one found in the search is selected, and logs with any other calculation are dropped from consideration.

- b. MAX
- c. MIN
- d. SUM
- e. SUM OF SQUARES
- f. STANDARD DEVIATION
- g. NUM_OF_VALUES

At this point the set of logs still under consideration have the same log period (closest to percent of coverage of requested time period), and calculation algorithm.

3. This set is now searched for the specified retrieval type. One log is selected according to retrieval type as follows:
 - a. If the retrieval type is RAW, one log is selected in the following order of preference:
 - 1) Pick the log whose storage interval is equal to the time between points in request.
 - 2) Pick the log whose storage interval is greater than and closest to the time between points (so request can be completed in one response).
 - 3) Pick the log whose storage interval is less than and closest to the time between points (more points exist than buffer can hold, MORE_DATA status is returned).
 - b. If the retrieval type is INTERPOLATED, one log is selected in the following order of preference:
 - 1) Pick the log whose storage interval is less than and closest to the time between points (so interpolation is more accurate).
 - 2) Pick the log whose storage interval is equal to the time between points in request.

- 3) Pick the log whose storage interval is greater than and closest to the time between points.
- c. If retrieval type is DISPLAY, one log is selected according to retrieval type as follows:
 - 1) Pick the log whose storage interval is equal to the time between points in request. The request is changed to RAW unless the log uses Deadband compaction, in which case the request is changed to INTERPOLATE/AVE.
 - 2) Pick the log whose storage interval is greater than and closest to the time between points. The request is changed to INTERPOLATE/AVE.
 - 3) Pick the log whose storage interval is less than and closest to the time between points. The request is changed to INTERPOLATE/AVE.

In addition to the seamless algorithm applied for retrieval requests, the following criteria have been added to better handle selection of a log for a request by access name, when logs are in a dual configuration:

- Uptime of node where log exists
- local is chosen over remote (local should always be faster than remote)
- sequence number of log. (with all other conditions equal, the lowest sequence numbered log is used. Sequence number is the '-1-o' or '-2-o' attached to the generated log name.

3.7.7 Queries for Messages

Use the EAMSGLOG (Master) to read history messages for AdvaInform Basic (User) objects.

Use the MODMSGLOG view to read historical messages related to CCF and TCL (MOD 300 only). These messages include:

- CCF alarm and parameter change messages
- TCL Unit and event messages
- Console log on and alarm acknowledgment messages

One message log stores all CCF and TCL messages, so you are not required to provide the log name in the query. The principle attributes in MODMSGLOG are listed in [Table 3-4](#).

Table 3-4. MODMSGLOG Attributes

Column Name	Format	Description
IDX:	NUMBER	Internal index column.
SOURCE_NODE	NUMBER	Node address message came from.
NETWORK:	NUMBER	Subdevice
LOCALTIME	DATE	Time of the message (one second resolution).
UTC	NUMBER	utc of the message in seconds.
MICROSECONDS	NUMBER	Micro second portion of time of the msg.
MESSAGE_TYPE	NUMBER	Type of message: 0 = log on/log off messages 1 = alarm messages 2 = billboard messages 3 = parameter change messages 4 = ladder logic messages 5 = Statistical Process Control (SPC) messages 6 = Alarm acknowledgment messages 7 = Diagnostic messages 8 = Batch 300 messages
MESSAGE_COUNT	NUMBER	Not implemented.
PRIORITY	NUMBER	Not implemented.
UNIT_NAME	CHAR(12)	Unit message originated from.
TAG_NAME	CHAR(32)	Tag message is for.
BATCH_NAME	CHAR(40)	Batch message is associated with.
SEQUENCE_NAME	CHAR(12)	Sequence that generated the message.
MESSAGE_STATUS	NUMBER	Status of the message entry.
MESSAGE	CHAR(255)	The message text.

3.7.7.1 Basic Query for Historical Messages

This query returns all messages in MODMSGLOG.

Query

```
SELECT tag_name, message
FROM MODMSGLOG
```

3.7.7.2 Query for Historical Messages in Time Range

This query returns all messages that occurred after a specified time.

Query

```
SELECT tag_name, message
FROM MODMSGLOG
WHERE to_date(localtime, 'yyyy-mm-dd hh24.mi.ss')>=
to_date('03/15/96 08:00:00', 'mm/dd/yy hh24:mi:ss')
```

3.7.7.3 Query for a Specific Type of Historical Message

This query returns all alarm messages in MODMSGLOG.

Query

```
SELECT tag_name, message
FROM MODMSGLOG
WHERE message_type = 1
```

3.7.7.4 Combine Message Type and Time Range

This query returns all parameter_change messages that occurred in a specified time range.

Query

```
SELECT tag_name, message
FROM MODMSGLOG
WHERE message_type = 3
AND localtime >= 1996-03-12 00.00.00'
AND localtime <= 1996-03-15 00.00.00'
```

3.7.7.5 Query for a Message with a Specified Text String

This query returns all messages that include the text string “operator”.

Query

```
SELECT tag_name, message
FROM MODMSGLOG
WHERE message LIKE '%OPERATOR%'
```

TIP

The % character is a wildcard character that represents any number of characters. The underscore (_) character represents a single character.

NOTE

Use wildcards (_, %) to specify the names of object instances only when necessary. It is generally better to use the full names as you configured them.

3.7.8 How to Read Basic and User Objects

Basic and User Objects in the local Advant Station can be read using either AdvaInform SQL*Connect to query objects, or direct SQL queries against user objects. CO_xx_INST tables in the AdvaInform relational database.

Each Basic and User Object type has a corresponding userobjects.CO_xx_INST table. The xx in the table name is the object type name. For example, to read an AO object, the query is in the format

```
SELECT value FROM userobjects.co_ao_inst WHERE name =  
'ao_five';
```

Reading an object instance from a userobjects.CO_xx_INST table takes approximately the same access time as using AdvaInform SQL*Connect. The attributes in the two tables are the same; however, the information in the userobjects table may be old (since it is possible to turn off shadowing of attribute information from the user objects). Therefore, you should read object information using SQL*Connect. It is recommended that you add the Basic Object and User Object instances to the location table.

The only method for reading Basic and User Objects in remote Advant Stations is via AdvaInform SQL*Connect. The queries are written against the object type tables. For example,

```
SELECT value FROM ao WHERE name = 'ao_five';
```

3.7.9 How to Join Data from Two Tables in a Single Query

In some cases, you may want to combine the data from two tables or views in a single query. For example, the following query retrieves the timestamp and entryvalue attributes from the numlogval view for all instances where the entryvalue is greater than the high limit. In order to compare the entryvalue to the high limit, the hi_limit attribute must be retrieved from the numlogconfig view. This view provides access to all configuration parameters for numeric log objects.

The way to join the two views is to use an attribute that is common to both, and set these attributes equal to each other in the WHERE clause.

Query

```
SELECT to_char(time, 'mm/dd/yy hh24:mi:ss'), timestamp,  
entryvalue  
FROM numlogval, numlogconfig  
WHERE numlogval.name = 'F_OIL,MEASURE'  
AND numlogval.name = numlogconfig.name  
AND entryvalue > hi_limit
```

A more common application of joins is in querying for PDL data, where you may require data for a single batch, and that data must be accessed from more than one view. Refer to the *PDL User's Guide* for examples.

Chapter 4 Runtime Operation

4.1 Product Operation

A number of Reports operations take place automatically during runtime without requiring intervention from the operator. Chief among these operations is report scheduling. It is set up when you configure the report object instances. The report automatically executes whenever the current time satisfies a condition as defined in the report's configuration, and cyclic scheduling is enabled. For example, one report may be configured to execute everyday at 16:00, another on the last day of the month, and so on.

The finished reports from these cyclic executions automatically go to the printers, ASCII files, pdf files, html files, and historical logs as defined in the report's parameter lists.

Multiple reports can execute at the same time.

If an error occurs during execution you are notified via a system error message.

4.2 Operating Overview

4.2.1 Operator Functions Supported

During runtime an operator at a terminal can:

- read and manipulate the report schedule. When you manipulate a schedule you can:
 - start cyclic scheduling of reports.
 - stop cyclic scheduling of reports.
- enable or disable cyclic, program, event (Master only), or manual scheduling of reports.
- manually schedule report execution.
- manipulate finished reports in the historical storage. When you manipulate a finished report you can:
 - retrieve reports from history for viewing on the screen or printing on printers.
 - modify character-based reports and save the modified versions to history.
 - Archive reports to cassette tape using history software procedures.

NOTE

You can save a report to a file in ASCII, postscript, PDF, or HTML format. Reports stored in such files are outside the AdvaInform Reports domain. Any manipulation you can normally perform on a file in the specified format can be performed on the report file. AdvaInform Reports has no control, and therefore no responsibility, for any modifications done to the file. A report cannot be returned to AdvaInform Reports from the file.

4.2.2 Operator Interface

Runtime interaction with the package is performed via the View Logs, Manual Execution, and Report Status windows. You can access these windows from the IMS Menu, from AdvaCommand display menus, and from menu bars of the various windows.

4.3 Runtime Tutorial

This section is not applicable.

4.4 Operating Instructions

4.4.1 How to View the Report Status

The Report Status window, [Figure 4-1](#), lists the reports in the system, shows their status, and allows you to start and stop cyclic report scheduling.

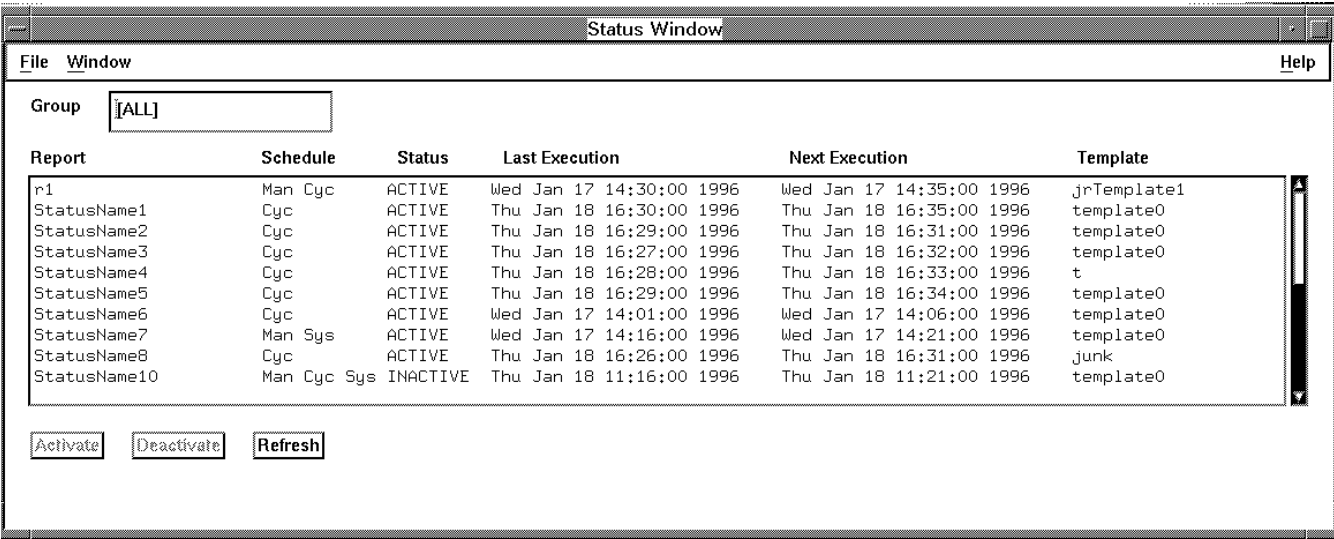


Figure 4-1. Report Status Window

The window is automatically updated every ninety seconds. You can access the Report Status window via the IMS Menu, or via AdvaCommand windows. The information presented in this window is described below.

Report	Report Name
Schedule	Indicates whether cyclic scheduling is active or inactive. Sys - report is being scheduled under system control (cyclic or on event)

	Man - report is for manual control
	Pro - report is to be executed by AdvaInform User API control or REPORT statement in a TCL program
Status	Report statuses include: Running - report is executing or available for scheduling Stopped - report is not executing or available for scheduling
Last Execution	Time of previous execution
Next Execution	Next scheduled execution
Template	Oracle Reports template for the report
Activate	Puts the selected report into the cyclic schedule. The Scheduling fields are ON after this button is selected. The report to be activated must have cyclic scheduling enabled on its Report Configuration window.
Deactivate	Removes the selected report from the cyclic schedule. The Scheduling fields are OFF after this button is selected.

4.4.2 How to Activate or Deactivate Scheduling for a Report

System scheduling for a report can be activated and deactivated from the Report Status window. This is performed by using the **Activate** and **Deactivate** buttons. For example, to put a report on the schedule:

1. Select one or more reports to be activated.
2. Select the **Activate** button. The Scheduling Field entry changes to : ON.

The reports then execute according to the conditions defined via its condition list. The report object instances must have scheduling enabled as described in [Section 4.4.3, How to Enable or Disable Report Scheduling Modes](#).

The **Deactivate** button is used in a similar fashion to take a report off the report schedule. That should be done for example if the report is about to be modified.

4.4.3 How to Enable or Disable Report Scheduling Modes

Manual and system scheduling of a report object instance can be enabled or disabled from the Report Configuration window. To enable or disable these scheduling modes:

1. Open the Report Configurator window.
2. Open the desired report object instance as follows:
 - a. Choose **Open** from the **File** menu.

This displays the Open Report dialog box which shows the groups that have report object instances defined for them, [Figure 4-2](#).

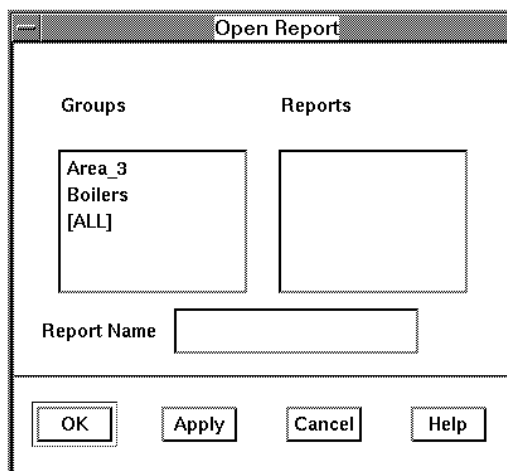


Figure 4-2. Open Report Dialog Box

- b. Select the group whose report object instance you want to open, [Figure 4-3](#).

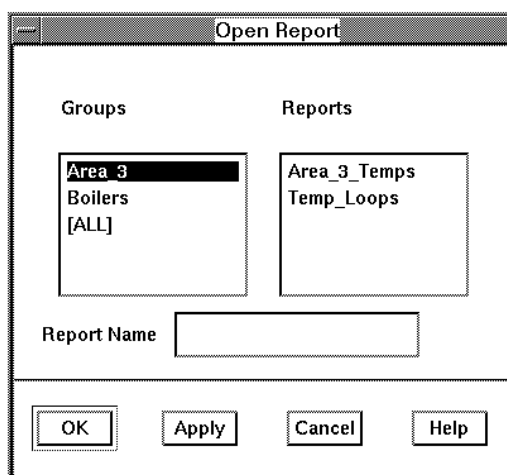


Figure 4-3. Reports Shown for Selected Group

When you select the group, the report object instances for that group are displayed in the Reports box.

- c. Select the report you want to open and then click on **OK**.

This displays the configuration data for the selected report object instance in the Report Configuration window, [Figure 4-4](#).

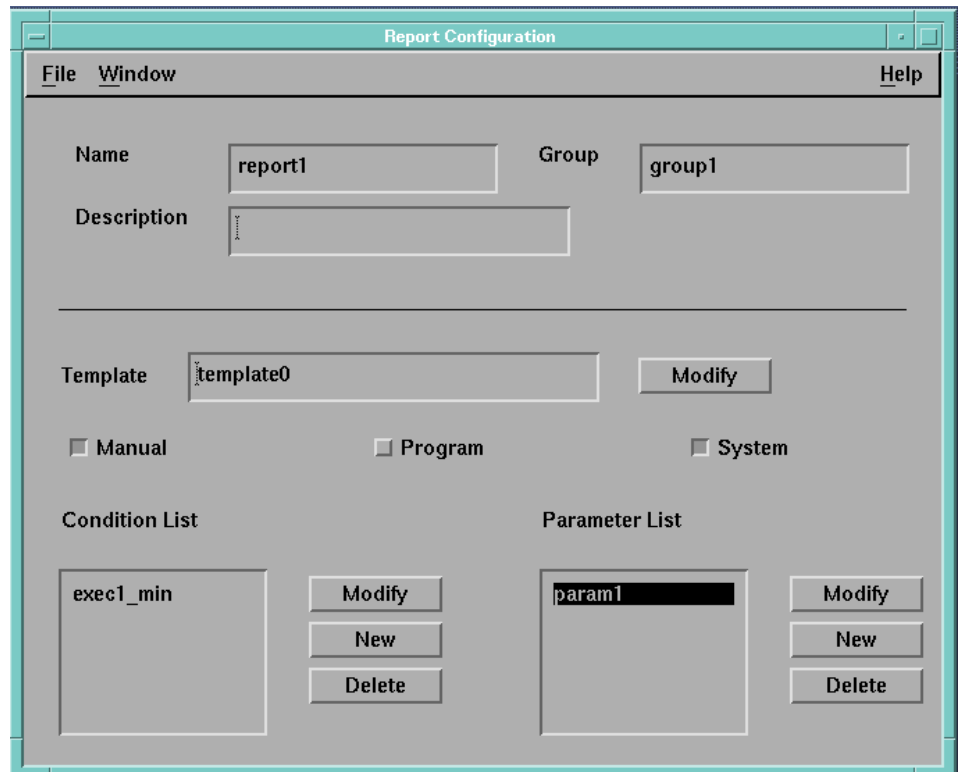


Figure 4-4. Report Configuration Window

3. Specify the scheduling options.

There are three scheduling modes: Manual, Program, and System. You can enable one or any combination of these modes by selecting the corresponding button, [Figure 4-5](#).

☒ **Manual** ☐ **Program** ☒ **System**

Figure 4-5. Enable Buttons for Scheduling Modes

Manual - allows scheduling by operator command from the Manual Report Execution window.

Program - allows scheduling via commands from a program via AdvaInform User API, or the REPORT statement in a TCL program.

System - allows scheduling according to conditions specified in the Report Configuration window.

4. To save the changes, choose either **Save** or **Save As** from the **File** Menu.

4.4.4 Manual Execution

You can execute a report manually via the Manual Execution window, [Figure 4-6](#).

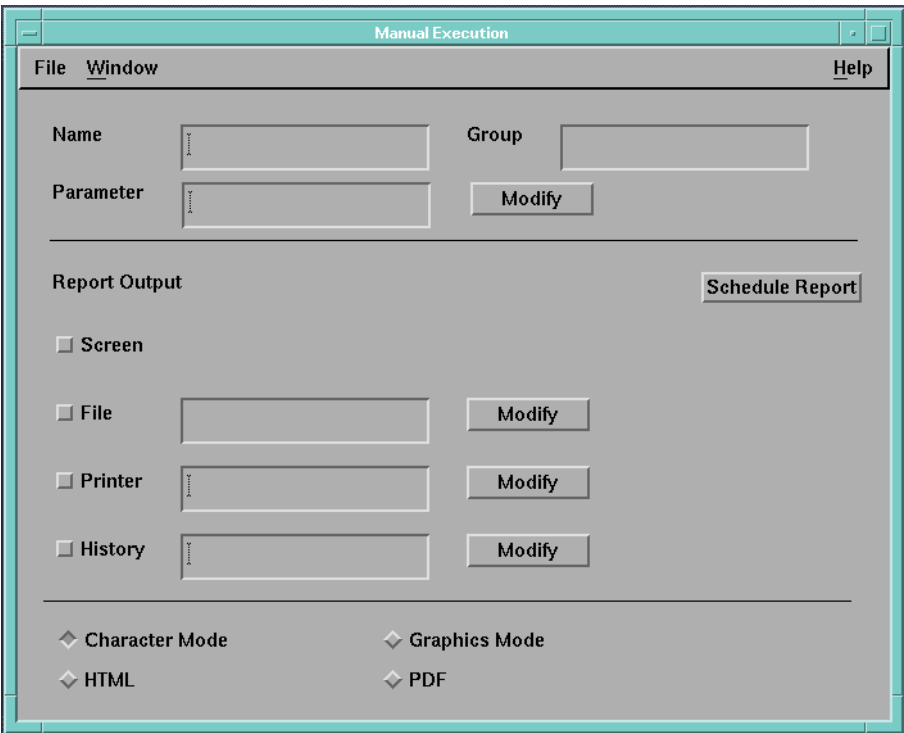


Figure 4-6. Manual Execution Window

The fields/buttons in this window are described below:

Name	This field displays the name of the report object instance.
Group	This field displays the group that contains the report object instance.
Parameter	This field displays the name of the Parameter List to use when the report executes. Enter the name directly, or click on Modify to display a selection box for parameter lists. Note that for manual execution, only the report template execution parameters are used from this list. Execution parameters for report output destination are specified below.

Screen	Click on this button to send the finished report to the screen currently being viewed. Character mode reports do not require third-party applications to be viewed on the screen. HTML reports do require a third party application for viewing. If you need the ability to launch third-party print routines, you must enter the appropriate command lines in the directory: <pre>/home/opt/advant/Reports/user.environment</pre>
File	Use this field to specify a file to send the finished report to. The file format can be Character (ASCII), postscript, PDF, or HTML as specified by the chosen report output mode. To enable the File parameter click on the File check box. Then click on the Modify button to display a dialog box for specifying the file, Figure 3-86. The full path is required. Use the file naming conventions of your operating system. You can not enter the file path directly in this field. You must enter it via the dialog box. NOTE: If you intend to send report output to a PDF or HTML file, it is recommended that the file name extension match the file type (for example .htm or .html for HTML files, or .pdf for PDF files). This is because some remote browsers are not able to display the files properly without these extensions. Once stored in a file, the report outside the AdvaInform Reports domain. There are no controls over the modifications that can be done to the file. A report cannot be returned to AdvaInform Reports from the file.
Printer	Use this field to specify a printer to send the finished report to. To enable the Printer parameter click on the Printer check box. Then click on the Modify Button to display a dialog for selecting a printer, or enter the printer name directly if you know the printer name. The printer name is case-sensitive. Character, PostScript, and PDF mode reports do not require third-party applications to be printed. HTML reports require a third party application for printing. If you need the ability to launch third-party print routines, you must enter the appropriate command lines in the directory: <pre>/home/opt/advant/Reports/user.environment</pre>
History	Use this field to specify a report log to send the finished report to. The file format can be Character, postscript, PDF, or HTML as specified by the chosen report output mode. To enable the History parameter click on the History check box, then enter the access name for the report log as defined in AdvaInform History. A check will be done that the log really exists.
Schedule Report	This button starts the report execution.

Character Mode	You can select any combination of report output destinations for character mode reports. Character mode reports do not require third-party applications to be viewed on the screen or printed. Note: Report templates that are saved as Graphic type may not run properly in this mode.
PostScript Mode	You can select any combination of report output destinations for graphic mode reports, with the exception of screen output. You can not select screen output in combination with any of the other report output destinations.
HTML	You can select any combination of report output destinations for HTML reports. Reports stored in HTML format can be made available to PC clients equipped with web browsers. These files can be manipulated in the same ways that other HTML files can be manipulated. HTML reports require third party applications for both viewing and printing. If you need the ability to launch third-party viewers and print routines, you must enter the appropriate command lines in the directory: <code>/home/opt/advant/Reports/user.environment</code>
PDF	You can select any combination of report output destinations for PDF reports. Reports stored in PDF format can be made available to PC clients equipped with web browsers. These files can be manipulated in the same ways that other PDF files can be manipulated. Adobe Acrobat Reader which supports viewing and printing PDF files is included with AdvaInform Reports software.

To manually execute a report:

1. Open the Manual Execution window.
You can open this window via any one of the following methods:
 - Choose **Manual Execution** from the **Window** menu in the Report Configuration window, or from the **Report** submenu in the **AdvaInform** menu in IMS Menu.
 - An AdvaCommand window.
2. Open the specific report object instance to be executed via the **Open** menu item in the **File** menu.
This displays the configuration data for the selected report object instance in the Manual Report Execution window.
3. Select a parameter list to be used with the execution
Enter the parameter list directly if you know the name, or click on **Modify** to display a menu of parameter lists and select one from the menu. If you want to specify no parameter list, choose the DEFAULT option.
4. Select the Report Output (destinations). You can select one or more destinations in any combination.

- a. If you want to send the finished report to the screen you are now viewing, select the **Screen** Button.
- b. If you want to send the finished report to a file, first select the **File** button to enable the option, then click on **Modify** to open the File Selection dialog. How to specify the file is described further in [Section 3.6.2.1, Parameter List Configuration Dialog](#).
- c. If you want to send the finished report to a printer, first select the **Printer** button to enable the Printer, and then enter the Printer name in the Printer field. If you know the name of the printer, enter the name directly; otherwise, you can click on **Modify** to display a list of printers.
- d. If you want to send the finished report to the historical storage, first select the **History** Button to enable the option, and then enter the access name of the report log where you want to store the report. You can use the **Modify** button to select an existing report log in History.

NOTE

Do not use the prefix (\$HS) or the suffix (-1-0) in the name. Also, the name is case sensitive.

5. Click on the **Schedule Report** Button to execute the report.

4.4.5 How to Manipulate Reports in Historical Storage

The AdvaInform History software provides long term storage for reports. History stores reports in report logs. You can perform the following operations on report logs:

- Retrieve reports from history for viewing on the screen or printing on printers
- Modify character-based reports and save the modified versions to history
- Retrieve report logs from history and save them to files

4.4.5.1 How to View a List of Report Logs

You can view a list of the report logs via the View Logs window, [Figure 4-7](#). This window can be opened via several methods. You can:

- choose **View Logs** from the **Window** menu in the Report Configuration, Manual Execution, or Report Status windows.
- choose **View Logs** from the **Report** submenu in the **AdvaInform** menu in IMS Menu.
- An AdvaCommand window.

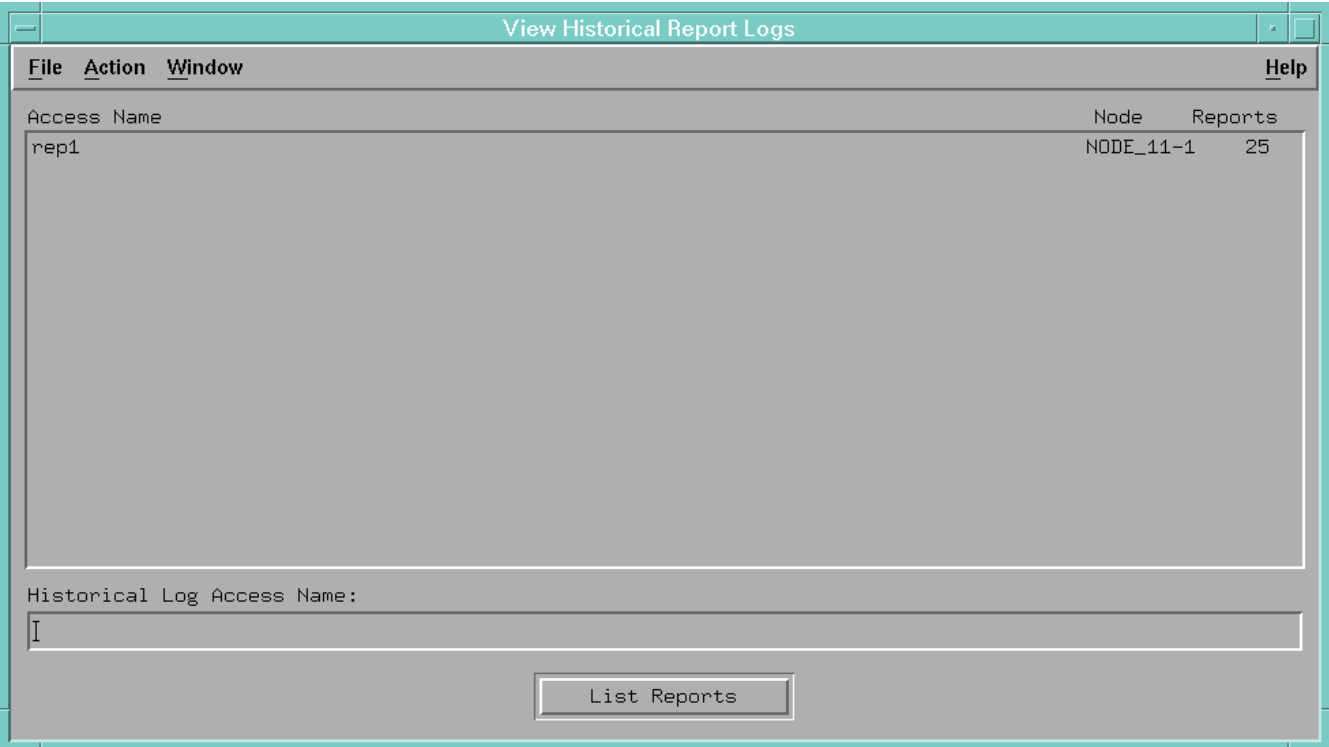


Figure 4-7. View Logs Window

The information presented in this window is described below.

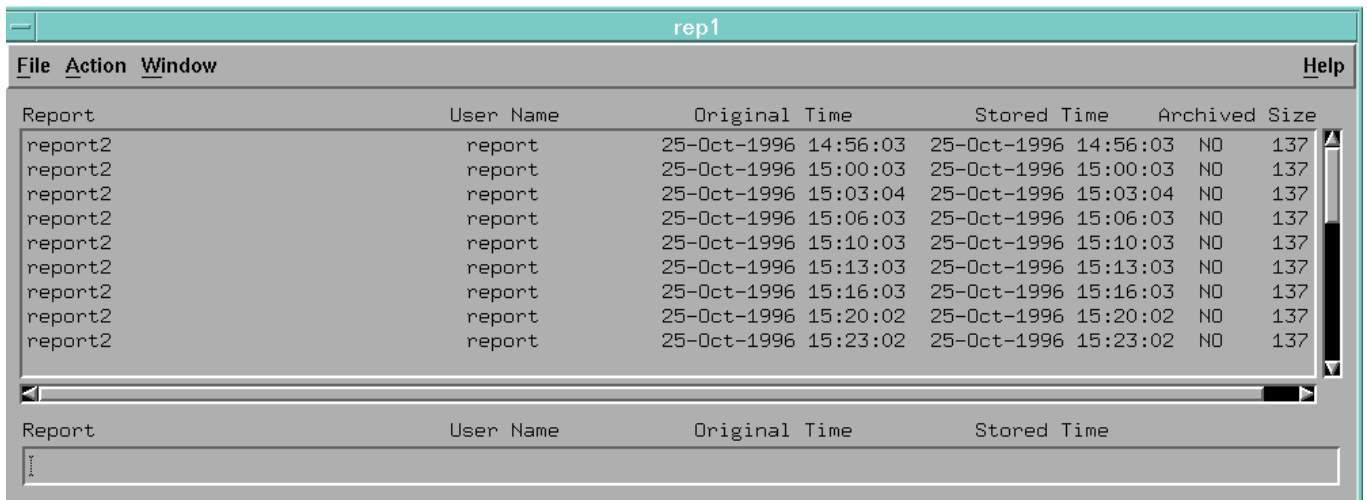
Access Name	Name of log as defined in Composite Log Window in History Builder.
Node	Location of log.
Reports	Number of reports in the log.
Access Name Box	When you select a log, its name is displayed in this box.
List Reports Button	This button opens the Report List window for the selected log.

4.4.5.2 How to View a List of Reports in a Report Log

To view a list of the reports in a report log:

1. Open the View Logs window via any one of the methods described in [Section 4.4.5.1, How to View a List of Report Logs](#).
2. Select the report log to be viewed.
3. Select the **List Reports** button at the bottom of the window.

This displays a list of reports in the report log, [Figure 4-8](#).



Report	User Name	Original Time	Stored Time	Archived	Size
report2	report	25-Oct-1996 14:56:03	25-Oct-1996 14:56:03	NO	137
report2	report	25-Oct-1996 15:00:03	25-Oct-1996 15:00:03	NO	137
report2	report	25-Oct-1996 15:03:04	25-Oct-1996 15:03:04	NO	137
report2	report	25-Oct-1996 15:06:03	25-Oct-1996 15:06:03	NO	137
report2	report	25-Oct-1996 15:10:03	25-Oct-1996 15:10:03	NO	137
report2	report	25-Oct-1996 15:13:03	25-Oct-1996 15:13:03	NO	137
report2	report	25-Oct-1996 15:16:03	25-Oct-1996 15:16:03	NO	137
report2	report	25-Oct-1996 15:20:02	25-Oct-1996 15:20:02	NO	137
report2	report	25-Oct-1996 15:23:02	25-Oct-1996 15:23:02	NO	137

Figure 4-8. Report List Example

Each report in the log is identified by the following:

Report	Name of the report.
User Name	User who configured the report.
Original Time	Execution time.
Stored Time	Time the report was saved to the log.
Archived	This column indicates whether or not the report log has been archived to tape.
Size	Size of the finished report in bytes.

4.4.5.3 How to View Reports

Character and PDF mode reports do not require a third-party application to be viewed on the screen. HTML reports do require a third party application for viewing. If you need the ability to launch third-party viewers, you must enter the appropriate command lines in the directory:
`/home/opt/advant/Reports/user.environment`

To view a report:

- 1. Open the Report List for the Log as described in [Section 4.4.5.2, How to View a List of Reports in a Report Log](#).
- 2. Select the desired report from the report list.
- 3. Choose **View** from the **Action** Menu.

For HTML reports, this launches the third-party viewer as specified in the command line. For PDF reports, this launches Adobe Acrobat Reader. For Character mode reports, this displays a submenu for selecting a text editor, [Figure 4-9](#).

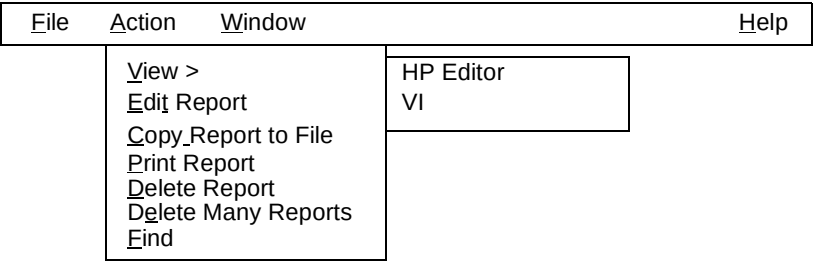


Figure 4-9. Submenu for Text Editors

Choose either **HP Editor** or **VI**. When you close the editor, control reverts to the window that previously had control.

4.4.5.4 How to Print Reports

The printer is added to the Advant Station as described in the *Advant Station IMS User's Guide*. When the printer is added, it must be made known to the node running the Reports software.

Character, PostScript, and PDF mode reports do not require third-party applications to be printed. HTML reports do require a third party application for printing. If you need the ability to launch third-party print routines, you must enter the appropriate command lines in the directory:

`/home/opt/advant/Reports/user.environment`

To print a report from a log:

- 1. Open the Report List for the report log that contains the report. See [Section 4.4.5.2, How to View a List of Reports in a Report Log](#).
- 2. Select the report to be printed.
- 3. Choose **Print Report** from the **Action** Menu. This displays a list of available printers.
- 4. Select a printer and then select the **Print** Button.

4.4.5.5 How to Modify Reports

To modify a report from a report log:

1. Open the Report List for the report log that contains the report. See [Section 4.4.5.2, How to View a List of Reports in a Report Log](#).
2. Select the report to be modified.
3. Choose **Edit Report** from the **Action** Menu.

This displays a submenu for selecting a text editor, [Figure 4-10](#).

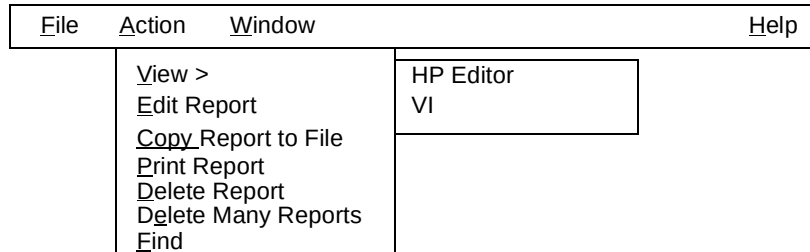


Figure 4-10. Submenu for Text Editors

4. Choose either **HP Editor** or **VI**.

This opens the report in the selected editor. Now you can:

- remove or change old text.
- add new text.
- add or delete lines and columns.
- modify field values.

An asterisk is displayed in each line you modify to indicate the line has been changed.

5. Save the report. This saves the modified copy to disk. The original copy remains on disk also.

The new report is automatically assigned a name that indicates it is a modified version of a finished report. For instance, if Boiler_1500 is the original report, when it is modified, the modified copy is saved as Boiler1500.1. If the original report is modified a second time, the result is Boiler1500.2. If Boiler_1500.1 is modified, the modification is saved as Boiler_1500.1.1.

When you close the editor, control reverts to the window that previously had control.

NOTE

You can not edit a report created in PostScript, HTML, or PDF format.

4.4.5.6 How to Delete a Report from a Report Log

A report can be removed from a report log as follows:

1. Open the Report List for the report log that contains the report. See [Section 4.4.5.2, How to View a List of Reports in a Report Log](#).
2. Select the report to be deleted.
3. Choose **Delete Report** from the **Action** Menu. This displays a prompt to verify whether or not you want to delete the report.
4. Click **OK** to verify and delete the report, or **Cancel** to cancel the delete operation.

4.4.5.7 How to Delete Many Reports from a Report Log

Many reports can be removed from a report log as follows:

1. Open the Report List for the report log that contains the reports. See [Section 4.4.5.2, How to View a List of Reports in a Report Log](#).
2. Choose **Delete Many Reports** from the **Action** Menu. This displays the Delete Reports dialog box, [Figure 4-11](#).

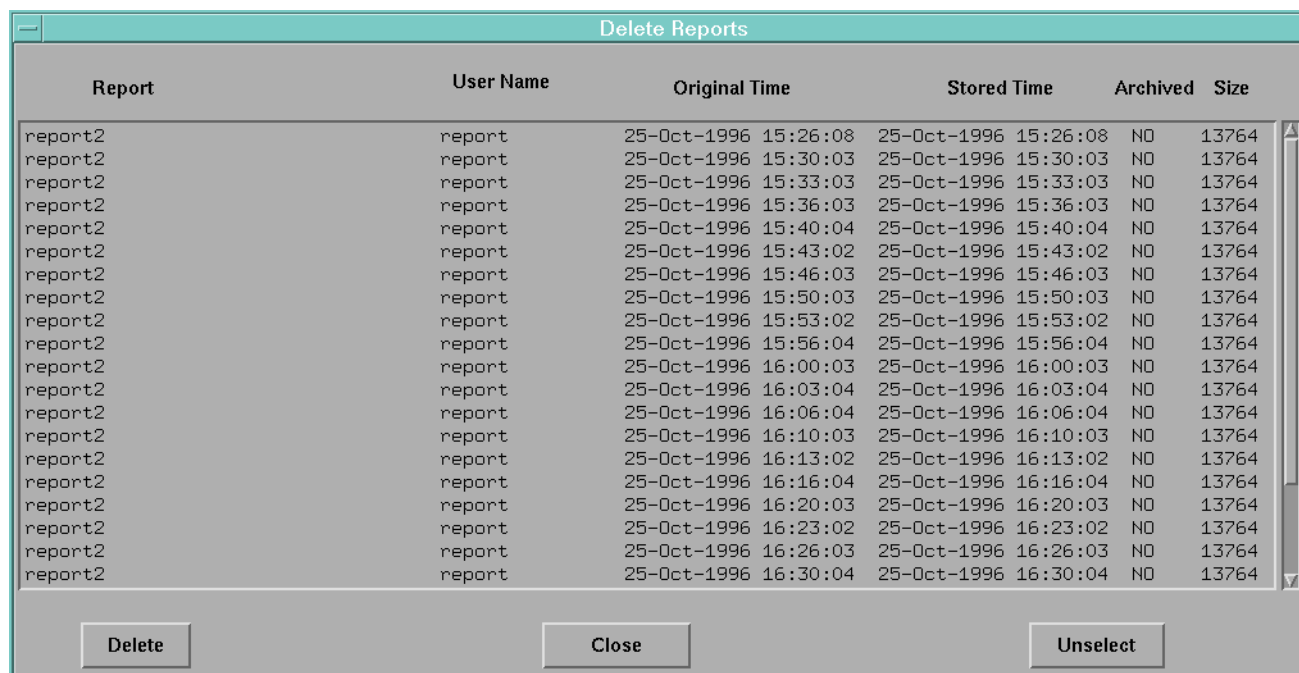


Figure 4-11. Delete Reports Dialog Box

3. Select reports to delete. To select multiple non-contiguous reports, hold down **<Ctrl>** as you select the reports. To select multiple reports in a row, select the first report, hold down **<Shift>**, and then select the last report. To unselect all selected reports, click **Unselect**.
4. Click on **Delete** when you are finished.

4.4.5.8 How to View a List of Restored Reports in a Report Log

To view a list of restored reports in a report log:

1. Open the View Logs window via any one of the methods described in [Section 4.4.5.1, How to View a List of Report Logs](#).
2. Select the report log to be viewed.
3. Choose **List Restored Reports** from the Action menu.

This displays a list of restored reports in the restored report log.

NOTE

All functions available in the Report List window are also available in the Restored Report List window, except that you can not modify restored character - based reports.

4.4.5.9 How to Search for Reports in Historical Storage

To find finished reports in the historical storage:

1. Choose **Find** from the **Action** Menu.

This displays a Find Report Dialog Box, [Figure 4-12](#).

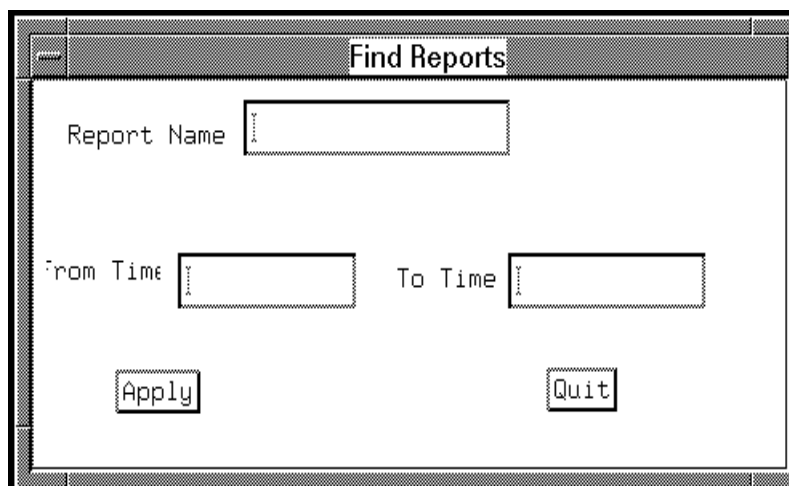


Figure 4-12. Find Report Dialog Box

2. Enter the Report name.
3. You have the option of entering a time interval for the requested reports via the From Time and To Time fields.

The format for the time entries must be the same as the format used on the Report List for the report. A North American report list uses the format dd-mon-yyyy hr:min, e.g.17-Sep-1993 14:20. If no time interval is specified, the software searches for all finished reports with the specified name.

4. Click on **Apply**.

Any reports that fit the search criteria are displayed in the Reports List.

4.4.5.10 Archiving Finished Reports

You can use the AdvaInform History software to put archive copies of finished reports onto DAT tapes.

This is performed on the report log level in the same manner that all history logs are archived. When you build a report log, you have the option of assigning it to an archive group. All logs in the archive group are archived to tape simultaneously either on a periodic basis or upon operator command. The AdvaInform History User's Guide explains archive groups.

4.4.5.11 How to Copy Report Logs to a File

You can retrieve a report log stored in any one of the four formats (Character, PostScript, PDF, or HTML) and save the report log to a file in the directory:

`/var/opt/advant/Reports`

To do this:

1. Open the Report List for the report log that contains the report. See [Section 4.4.5.2, How to View a List of Reports in a Report Log](#).
2. Select the report to be sent to a file.
3. Choose **Send Report to File** from the **Action** Menu.

This displays the Report Log Entry File Name dialog box, [Figure 4-13](#).

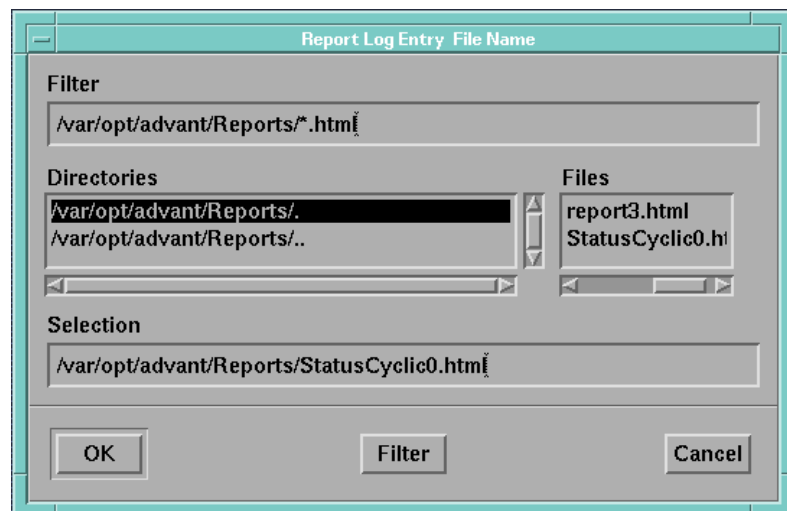


Figure 4-13. Report Log Entry File Name Dialog Box

The selection field in this dialog indicates the format of the selected log. For instance, in [Figure 4-13](#) the log name is `StatusCyclic0`, and the format is `html`.

4. Click **OK** to save the report log to the directory indicated in the Directories list.

4.4.6 How to Retrieve Encapsulated data

The encapsulated data can be selected from the ENCAP_DATA table. The ENCAP_DATA table has following columns, [Table 4-1](#).

Table 4-1. ENCAP_DATA Table Columns

Column name	Description	Data Type	Length
Access_Name	Report access name defined for the report in History output field	CHAR	81
Field_Name	Name of field that encapsulated data is retrieved from	CHAR	20
Execution_Time	Time when encapsulated data was stored.	CHAR	84
Segment_ID	Tag number of value from field specified to contain the encapsulated data.	NUMBER	
Report_Name	Name of the report instance	CHAR	65
Value	Value (or values) encapsulated for a report.	LONG	

These examples show how to select encapsulated values from a report log for a specific report. The queries in these examples are entered via Oracle Reports.

Example 1:

This query retrieves encapsulated data from report Test_Encap in REPORT_LOG1 after 12 o'clock on the 11:th of this month this year.

```
SELECT Field_name,Value FROM Encap_Data
WHERE Access_Name = 'REPORT_LOG1' AND
      Report_Name = 'Test_Encap' AND
      TO_DATE(Execution_Time, 'YY-MM-DD HH24:MI:SS')
      > TO_DATE('11 12:00:00', 'DD HH24:MI:SS')
```

Example 2:

This query retrieves encapsulated data from report Test_Encap in REPORT_LOG1 between 10 and 12 o'clock on the 11:th of this month this year.

```
SELECT Field_name,Value FROM Encap_Data
WHERE Access_Name = 'REPORT_LOG1' AND
      Report_Name = 'Test_Encap' AND
      Execution_Time > TO_DATE('11 10:00:00', 'DD HH24:MI:SS')
      AND Execution_Time < TO_DATE('11 12:00:00', 'DD
      HH24:MI:SS')
```

For more information about the TO_DATE function refer to Appendix A.

4.5 Runtime Operation Menus

4.5.1 Manual Report Execution Window Menus

This section describes the Manual Report Execution menu items, [Figure 4-14](#).

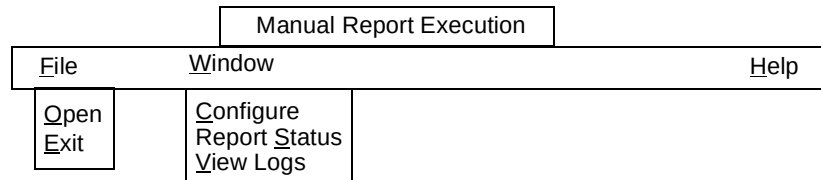


Figure 4-14. Manual Report Execution Menu Bar and All Menu Items

4.5.1.1 Manual Report Execution File Menu

File - Open

Use **Open** to display the configuration data for a report object instance in the Manual Execution window. When you choose **Open**, the open report dialog box is displayed. First select the group containing the report object instance, and then select the report object instance. Click on **OK** to remove the dialog box and display the configuration data in the Manual Execution window.

File - Exit

Exit closes the Manual Report Execution window and returns you to the previously opened window.

4.5.1.2 Manual Report Execution Window Menu

Window - Configure

Configure opens the Report Configuration window.

Window - View Logs

View Logs opens the View Logs window.

Window - Report Status

Report Status calls up the Report Status window.

4.5.2 View Logs Menus

This section describes the menus in the View Log window, [Figure 4-15](#).

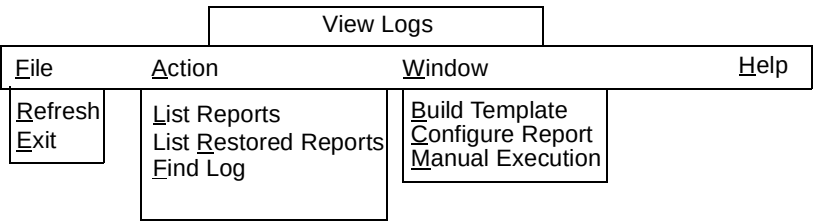


Figure 4-15. View Logs Menu Bar and All Menu Items

Note that the List Report Logs item is dimmed. That is, it is not a part of this version of reports.

4.5.2.1 View Logs File Menu

File - Refresh

Refresh updates the information in the View Logs window while a report is executing.

File - Exit

Exit closes the View Logs window and returns you to the previous window.

4.5.2.2 View Logs Action Menu

Action - List Report

List Report opens the Report List window which lists the reports of the currently selected log.

Action - List Restored Reports

List Restored Reports opens the Restored Report List window which lists the restored reports of the currently selected log.

Action - Find Log

Find Log opens the Find Log dialog box.

List Reports Logs creates a list of report logs in a particular node or to locate a specific log by name. A dialog box pops up when **List Reports** is chosen. Two text fields are provided to enter either the node name or the log name.

4.5.2.3 View Logs Window Menu

Window - Build Templates

Build Templates opens Oracle Reports.

Window - Configure Report

Configure Report opens the Report Configuration window.

Window - Manual Execution

Manual Execution opens the Manual Report Execution window.

4.5.3 Reports List Menus

This section describes the menu items in the Reports List window, [Figure 4-16](#).

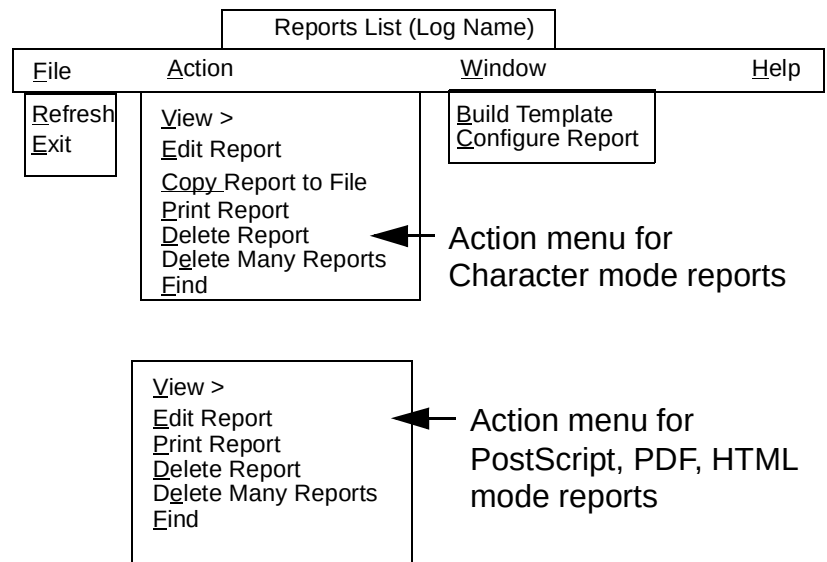


Figure 4-16. Reports List Menu Bar and All Menu Items

4.5.3.1 Reports List File Menu

File - Refresh

Refresh updates the report list information.

File - Exit

Exit closes the Report List window and returns control to the parent window (View Logs).

4.5.3.2 Reports List Action Menu

Action - View

For character mode reports, **View** displays a submenu for choosing the text editor to view the selected report. For HTML reports, **View** launches the third-party viewer as specified in the command line. For PDF reports, **View** launches Adobe Acrobat Reader.

Action - Print Report

Print Report displays a dialog box with a list of available printers. You choose one from the list to print the selected report.

Action - Edit Report

Edit Report displays a submenu for choosing the text editor to modify the selected report.

Action - Send Report to File

Send Report to File displays the Report Log Entry File Name dialog which is used to specify a file for storing the selected report log. These files are stored in the directory:

/var/opt/advant/Reports

Action Delete Report

Delete Report deletes a report. A dialog box is displayed to verify the delete command. Click **OK** to verify and delete, or click **Cancel** to cancel the command.

Action Delete Many Reports

Delete Many Reports displays the Delete Many Reports window for deleting more than one report at a time.

Action - Find

Find displays the Find Report dialog box. From this box you search for finished reports by entering the report name in the report name field. You also have the option to designate a time frame for the requested finished reports.

4.5.3.3 Reports List Window Menu

Window - Build Templates

Build Templates opens Oracle Reports Object Navigator window.

Window - Configure Report

Configure Report opens the Report Configuration window.

Chapter 5 Maintenance

5.1 Preventive Maintenance

Not applicable.

5.2 Fault Finding and Repair

Not applicable.

5.3 Preventive Maintenance

5.3.1 Maintainability

If a problem occurs that can not be solved by yourself or your system manager, please contact your local ABB Industrial Systems representative. This includes problems and errors that you have identified as being related to the AdvaInform Basic Software, AdvaInform and AdvaBuild options, the operating system HP-UX, Oracle RDBMS or any Oracle tool.

When you report an PMR, please enclose all of the supplementary information related to the problem that you have. For example, it may be a description of changes you made in your system prior to the problem. It may also be printouts of error logs and hard copies of configuration displays. The more information you can supply the shorter time for a response.

5.3.2 The HP-UX kernel

You should not change the settings of the HP-UX kernel. Buffers, swap space, semaphores and so on have been set to insure that Advant Station has the capacity to handle AdvaInform functionality and still maintain the best possible performance.

Some third party software might require rebuild of the HP-UX kernel. The system manager may only increase a system parameter. In that case you may change the capacity and performance of all functions in Advant Station.

5.4 Hardware Indicators

Not Applicable

5.5 Error Messages

5.5.1 Error Messages for MOD 300 Applications

5.5.1.1 Where to Read Error Messages

As you use AdvaInform Reports, various informational and diagnostic messages are generated. To access a listing of these messages:

- On the Advant Station with AdvaInform software, open the System Messages window. To open this window, choose **System Messages** from the **Station** menu on the IMS Menu window. Refer to the AdvaInform Basic Functions User's Guide for instructions on how to use this window.
- On the Advant Station with AdvaCommand software, select the DIAG target in the upper right corner of your display. This action automatically takes you to the DIAGNOSTIC MESSAGES DISPLAY and presents a sentence or phrase to help explain the situation or error you have encountered.

If you require outside assistance for the successful resolution of a situation or problem, the diagnostic error messages contained in this section may help. When you contact service for help, it is important to provide them with as much information as possible concerning your problem. To obtain more information, change the way the diagnostic messages are displayed. Select the OPERATOR_MSGS target from the DIAGNOSTIC MESSAGES DISPLAY. Using the NEXT CHOICE key, toggle to and then select ALL MSGS—CODE. This will provide you with the AP code associated with the error that has occurred. Find the AP code within this section to determine the cause and effect of your problem as well as the action you should take next. Have this information on hand if you need to contact service for additional support.

NOTE

Please record all messages that are not found in this manual and report them to ABB. The information you record should contain the following:

- error message text copied exactly as it appears on the screen
- error message code copied exactly as it appears on the screen
- type of node that generated the diagnostic
- sequence of events leading up to the diagnostic message
- any other important data

Providing the above information allows for a more efficient response and accurate documentation.

5.5.1.2 Error Message Format

The diagnostic error messages are presented in the following format:

AP Error Code (Severity)	Message ID:
	Message Text:
	Data 1 (%D1):
	Data 2 (%D2):
	Text Data (%T2):
	Cause:
	Effect:
	Action:

AP Error Code	Four digit hexadecimal code. The first two digits (AP) are the application ID, the last two digits are the message number.
Severity	Information (I), Warning (W), Error (E), Fatal (F)
Message ID	Text message ID
Message Text	Appears when ALL_MESSAGES_TEXT is selected on the diagnostic display.
Data 1 (%D1)	Numerical data 1
Data 2 (%D2)	Numerical data 2
Text Data (%T)	Message specific text
Cause	Reason for message
Effect	Effect of condition on system
Action	What you can do to correct the error

5.5.1.3 Reports Diagnostic Messages

The Reports messages are broken into various sections, each section associated with a particular series of error codes. For example, Section 5.3.3.1, contains all error codes associated with the AP code A5 (Message Enums). Therefore, all error codes associated with the Message Enums application will be found in [Section , Reports Diagnostic Messages](#).

AP A5 - Message Enums

A501 W	Message ID:	RGNodata
	Message Text:	No data was found looking for %T. rg:%D1
	Data 1 (%D1):	AdvaInform Reports error code
	Data 2 (%D2):	OMF error code
	Text Data (%T):	
	Cause:	No data exists for the request
	Effect:	
	Action:	Verify that the selected items exist and try again

A502 W	Message ID:	RGFileOpenError
	Message Text:	Error opening file %T. HP error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	System error code
	Text Data (%T):	File name
	Cause:	Could not open or create a file
	Effect:	
A503 W	Action:	Check disk space and delete files if disk is full
	Message ID:	RGLogNoExist
	Message Text:	Log %T does not exist in system.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	Log name
	Cause:	Attempted to access a log that does not exist
A506 W	Effect:	
	Action:	Verify that the log name is spelled correctly
	Message ID:	RGAllocateError
	Message Text:	Error Allocating space for %T.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
A509 W	Cause:	Could not allocate memory
	Effect:	
	Action:	Verify that the system is not overloaded
	Message ID:	RGNoReportTemplet
	Message Text:	No report templet specified.
	Data 1 (%D1):	
	Data 2 (%D2):	
A511 W	Text Data (%T):	
	Cause:	No report template found
	Effect:	
	Action:	
	Message ID:	RGSqlrExecError
	Message Text:	
	Data 1 (%D1):	Could not start process that executes a report
	Data 2 (%D2):	Verify the report's configuration
	Text Data (%T):	
	Cause:	Some part of the report configuration is missing
	Effect:	
	Action:	Verify that the report is configured correctly

A512 W	Message ID:	RGOracleConnectError
	Message Text:	Oracle connect error. TWO_TASK = %T
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	The Oracle TWO TASK connect string used
	Cause:	Usually, the TWO TASK string is wrong
	Effect:	Cannot connect to Oracle
A513 W	Action:	
	Message ID:	RGCommitError
	Message Text:	Oracle commit error. Oracle error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	Error code Oracle uses
	Text Data (%T):	
	Cause:	
A514 W	Effect:	Last changes made are lost
	Action:	Terminate report application. Restart it and verify all changes were made. Re-edit a report if necessary and then save.
	Message ID:	RGInvalidConnectParam
	Message Text:	Invalid Oracle TT. TWO_TASK = %T
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	Oracle Two Task used
A515 W	Cause:	Got an invalid two task string
	Effect:	Can notconnect to Oracle
	Action:	Terminate the report application, and then re-start it
	Message ID:	RGBadCommandLine
	Message Text:	Bad command line passed to %T
	Data 1 (%D1):	
	Data 2 (%D2):	
A516 W	Text Data (%T):	
	Cause:	A bad command line was passed to rgrunrep or rgstart process
	Effect:	The process will not run
	Action:	
	Message ID:	RGDeletionError
	Message Text:	Error deleting report %T. Oracle error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	Oracle error
	Text Data (%T):	
	Cause:	Could not remove a report from the database
	Effect:	The report will still exist in the report list
	Action:	Try to delete the report again

A517 W	Message ID:	RGOracleError
	Message Text:	An Oracle error occurred. Oracle error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	Oracle error
	Text Data (%T):	
	Cause:	Error while accessing Oracle
	Effect:	The last operation that was attempted, failed
A519 W	Action:	
	Message ID:	RGDefaultNode
	Message Text:	No default node could be found
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	Could not get the default node name
A520 W	Effect:	
	Action:	
	Message ID:	RGNoMemory
	Message Text:	No memory left for allocation.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
A521 W	Cause:	No more system memory
	Effect:	Can not run applications
	Action:	Terminate the application and try again later
	Message ID:	RGFileError
	Message Text:	Error accessing file %T HP error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	System error number
A522 W	Text Data (%T):	
	Cause:	Could not access listed file
	Effect:	
	Action:	
	Message ID:	RGTimeCheckError
	Message Text:	Time/Date error: %T.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	Time and date string
	Cause:	User entered an invalid time and date value
	Effect:	
	Action:	Check the time and date format

A523 W	Message ID:	RGFileNoExist
	Message Text:	File %T does not exist.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	A file could not be read or accessed
	Effect:	If this error came from rgRunRep, then the report will not execute
A527 W	Action:	
	Message ID:	RGNoAllocSpace
	Message Text:	No space allocated for %T.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	No system memory
A528 W	Effect:	
	Action:	
	Message ID:	RGExecuteShellError
	Message Text:	Failed to execute %T. HP error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	HP system error
	Text Data (%T):	
A529 W	Cause:	Could not start the build templates process
	Effect:	Can not start Oracle Reports
	Action:	Check the system error
	Message ID:	RGNoCapabilitiesToAssign
	Message Text:	No capabilities to assign.
	Data 1 (%D1):	
	Data 2 (%D2):	
A530 W	Text Data (%T):	
	Cause:	
	Effect:	
	Action:	
	Message ID:	RGErrorCreatingCapability
	Message Text:	Error creating capability. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	
	Effect:	
	Action:	

A531 W	Message ID:	RGErrorCreatingObjview
	Message Text:	Error creating object view. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	
	Effect:	
A532 W	Action:	
	Message ID:	RGErrorDeletingCapability
	Message Text:	Error deleting capability. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	
A533 W	Effect:	
	Action:	
	Message ID:	RGErrorDeletingObjView
	Message Text:	Error deleting object view. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
A534 W	Cause:	
	Effect:	
	Action:	
	Message ID:	RGErrorAssigningObjView
	Message Text:	Error assigning object view. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	
A535 W	Text Data (%T):	
	Cause:	
	Effect:	
	Action:	
	Message ID:	RGTranslateFailure
	Message Text:	Failed to translate %T. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	
	Effect:	
	Action:	

A536 W	Message ID:	RGOmfsyncDevError
	Message Text:	Error creating a sync device. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	
	Effect:	
A537 W	Action:	
	Message ID:	RGOmfsyncOpenError
	Message Text:	Error opening OMF. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error
	Text Data (%T):	
	Cause:	Could not open OMF
A538 W	Effect:	
	Action:	
	Message ID:	RGContextError
	Message Text:	Error creating OMF context. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error
	Text Data (%T):	
A539 W	Cause:	Could not open OMF
	Effect:	
	Action:	Check History log name
	Message ID:	RGMonitorTimeout
	Message Text:	Monitor timeout. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error
A540 W	Text Data (%T):	
	Cause:	Monitor timed out
	Effect:	
	Action:	
	Message ID:	RGNoBaseCapability
	Message Text:	No base cap to reference. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error
	Text Data (%T):	
	Cause:	
	Effect:	
	Action:	

A541 W	Message ID:	RGSubmissionFail
	Message Text:	OMF submission failed. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error
	Text Data (%T):	
	Cause:	A submit request to OMF failed
	Effect:	
A542 W	Action:	
	Message ID:	RGInvalidData
	Message Text:	Invalid data received. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error
	Text Data (%T):	
	Cause:	Invalid data recieved from OMF
A543 W	Effect:	
	Action:	
	Message ID:	RGOmfCloseFailed
	Message Text:	Error closing OMF. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error
	Text Data (%T):	
A544 W	Cause:	Could not close OMF
	Effect:	
	Action:	
	Message ID:	RGOmfInitFailed
	Message Text:	Failed to initialize OMF.
	Data 1 (%D1):	
	Data 2 (%D2):	
A545 W	Text Data (%T):	
	Cause:	
	Effect:	
	Action:	
	Message ID:	RGCapabilityDataError
	Message Text:	Bad data from capability. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error code
	Text Data (%T):	
	Cause:	
	Effect:	
	Action:	

A547 W	Message ID:	RGOacleNodeByLogErr
	Message Text:	No oracle TT for log %T.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	Log name
	Cause:	Could not get the Oracle Two Task for the log
	Effect:	
	Action:	Verify that the log name is correct
A548 W	Message ID:	RGFileStatError
	Message Text:	Stat of file %T not avail.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	File is either not on the disk, or busy
	Effect:	
	Action:	Make sure the file is on the disk and has permissions set correctly
A549 W	Message ID:	RGSystemCommandFailed
	Message Text:	System call to '%T' failed.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	
	Cause:	Could not make a system call
	Effect:	
	Action:	
A550 W	Message ID:	RGErrorDeletingContext
	Message Text:	Error deleting OMF context. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error code
	Text Data (%T):	
	Cause:	
	Effect:	
	Action:	
A551 W	Message ID:	RGErrorDeletingMontor
	Message Text:	Error deleting OMF monitor. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error code
	Text Data (%T):	
	Cause:	The OMF monitor could not be deleted
	Effect:	
	Action:	

A552 W	Message ID:	RGErrorTerminatingMonitor
	Message Text:	Error terminating OMF monitor. OMF error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	OMF error
	Text Data (%T):	
	Cause:	OMF monitor could not be terminated
	Effect:	
A553 W	Action:	
	Message ID:	RGCommandTooLong
	Message Text:	Exec command for %T too long.
	Data 1 (%D1):	
	Data 2 (%D2):	
	Text Data (%T):	Name of the function to be started
	Cause:	System could not start the process
A554 W	Effect:	
	Action:	
	Message ID:	RGReportNotFound
	Message Text:	Report %T not found. Oracle error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	Oracle error
	Text Data (%T):	Report name
A555 W	Cause:	The report does not exist, or is on another node
	Effect:	
	Action:	Check the spelling of the report name. Make sure the correct node is being accessed
	Message ID:	RGDirectoryError
	Message Text:	Failed to %To directory. HP error = %D2
	Data 1 (%D1):	
	Data 2 (%D2):	System error number
	Text Data (%T):	Operation to be performed
	Cause:	The director does not exist, or has the wrong permissions
	Effect:	
	Action:	Make sure that 'usr/spool/lp/member' exists and that the permissions are correct

5.5.2 Error Messages for Master Applications

The error messages generated by AdvaInform Reports can be viewed in different ways - for example only those having a certain severity code - by using the features of the Error Log display (refer to *Advant Station 500 IMS User's Guide* for a description).

The error messages issued by AdvaInform Reports have the following standardized format:

Error string <P1><P2><P3>

- Parameters:** P1: description of parameter 1
P2: description of parameter 2
P3: description of parameter 3
The parameters are optional in an error message. This means that there might be no parameter, just a text, or one, two or three parameters.
- Audience:** The audience of the error message. Valid entries are Operator, Engineer, Service and Debug.
- Cause:** Cause of the error.
- Action:** Action to take when the error occurs.

The error messages are sorted by their severity in the following classes

- Fatal error messages
- Error messages
- Warning messages
- Information messages

5.5.2.1 Messages Box

The other types of error (or informational) messages that are issued by AdvaInform Reports are presented directly to the operator. Normally, the reason for the message is that the operator has entered an invalid command. In most cases, the operator himself can take the necessary actions to correct the command.

These types of messages are presented in a standardized message box, [Figure 5-1](#).

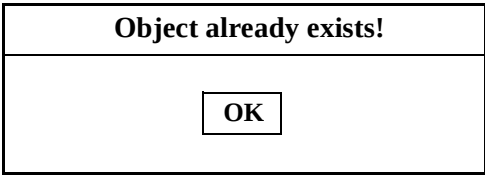


Figure 5-1. Example of an Error Message Dialog Box

The error message must be acknowledged (click on **OK**) before you are allowed to continue.
The error messages are self explanatory and are thus not described in detail here.

5.5.2.2 Reports Diagnostic Messages

Run time environment errors are reported as error logs. The 'process name' in the error log is always the name of the object that sent the error log. You study the error logs from the Error log display which is available from the IMS menu.

The error logs are sorted in two categories: **Fatal** and **Error**. The Fatal errors normally result in a termination of a process.

A listing of error logs, along with an explanation of each error log and a recommended course of action, follows. The error logs are documented as follows:

Error Text with parameters embedded in ‘ ’.

Parameters: Explanation of the parameters. If no parameters exist this section is left out.

Audience: The audience of the error message. Valid entries are Operator, Engineer, Service and Debug. Only presented if Audience is other than Operator or Engineer.

Cause: Cause of the error.

Action: Action to take.

5.5.2.3 Fatal error messages

Configuration error: Bad object reference

Cause: The connected object does not exist any more.

Action: Reconfigure the connection to a valid object.

Object Error: Instance table error ‘&object.name

Parameters: P1: Name of object for which the error occurred.

Audience: Service

Cause: Error when accessing the instance table. This error should not occur.

Action: Try to start the object once more. If the error still occurs, contact ABB.

Object error: OMF bind error ‘&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: Internal core system error, can affect future execution.

Action: Contact ABB.

Object error: Operation exception ‘&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: A run time error in the object execution occurred, for example, divide by zero. The object status bits error is set and the object is stopped.

Action: The object must be reconfigured to run again. It probably needs corrections in the object type implementation as well. If it is a basic object, report the error to ABB.

Object error: Reduce group error '&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: Internal core system error, can affect future execution.

Action: Contact ABB.

5.5.2.4 Category Error

General Errors

Connection error: Bad object reference '&connection.name &reference

Parameters: P1: Name of object for which the error occurred.
P2: Reference of object for which the error occurred.

Cause: The connected object does not exist any more.

Action: Reconfigure the connection to a valid object.

Error: No more free memory

Cause: No more free memory in the machine, no more room to create new objects.

Action: Delete a few objects handled by that Object Handler. If Advant Station 500, extend the UNIX kernel to accommodate larger programs. If Advant Station 800, extend the Object Handler's memory quotas.

Object error: Already active error '&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: Attempt to start an already started object. Probably two applications that "collide", one trying to activate an already active object. Perform an investigation which applications that access the object in question.

Action: Try to find out which applications interfere with one another.

Object error: Connection read error '&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: The connected object does not exist any more.

Action: Reconfigure the connection to a valid object.

Object error: Cyclic Normal Operation error '&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: The cyclic operation resulted in a fatal error.

Action: Check for additional error logs for more information on the fault. Reconfigure the object if that corrects the error. There may be errors in the object implementation as well.

Object error: Cyclic overflow error '&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: Cyclic normal operation overflow error. This means that the previous execution is not completed when a new request for cyclic execution is issued.

Action: Reconfigure the execution schedule. Raising the Object Handler priority (use the Object Type Builder) or lowering the priority of other programs may also solve the problem.

Object error: No network address '&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: No network address can be obtained.

Action: Check with the main menu System->Communication set up. If not set. Define it. Restart the Object handler If other problem. Contact ABB.

Object error: Trigger Normal Operation error '&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: A triggered (event-driven) execution of the Normal Operation resulted in a fatal error.

Action: Check for additional error logs for more information on the fault. Reconfigure the object if that corrects the error. There may be errors in the object implementation as well.

Table Space is full

Cause: No more free disk memory in the User Object Oracle disk file

Action: Contact your system manager. He needs to extend the available table space by using the Oracle DBA command Alter Tablespace. He is recommended to add a file of 1 MB to the table space IMS_USEROBJECTS. The disk file shall be placed under:

/disk1/products/data/standard/UserObjects/oracle/

Internal Software Errors

Error: Object state processing

Cause: Internal software error which should not occur. The object is stopped.

Action: Restart the Object Handler. Report the error to ABB.

Lower level error: & Object name

Parameters: P1: Name of connection/attribute for which the error occurred.

Cause: This error log is logged together with another log that describes the error more thoroughly.

Action: Look for another error log issued at the same time and take appropriate actions according to that.

Object error: OMF unbind error '&object.name and deactivate the object

Parameters: P1: Name of object for which the error occurred.

Cause: Internal software error which should not occur. The object is stopped.

Action: Restart the Object Handler. Report the error to ABB.

Object error: OMF view creation error '&object.name

Parameters: P1: Name of object for which the error occurred.

Cause: Internal software error which should not occur. The object is stopped.

Action: Restart the Object Handler. Report the error to ABB.

Property error: Lost reference'&property.name

Parameters: P1: Name of connection/attribute for which the error occurred.

Audience: Service

Cause: Internal software error which should not occur. The object is stopped. The message is combined with another error log.

Action: Restart the Object Handler. Report the error to ABB.

5.6 Frequently Asked Questions and Problems

Report is not working

If a report is not working, first verify that the Advant Station is connected to the TCP/IP network.

Report instances crash after change of language

The TO_DATE and TO_CHAR calls must contain the second parameter 'fmt', format, to make language switches possible.

If there are more than 50 select statements in a report towards SQL*Connect (for example NUMLOGVAL), the report is not generated.

The SQL*Connect is a separate Oracle database. You can not change, for example, open_cursors in the normal initims.ora configuration file. Instead you have to set for example, open_cursors=100 in the file \$ORACLE_HOME/abb/admin/initgtw.ora, and restart the Advant Station, including Oracle.

Appendix A SQL Statement Reference for AdvaInform Reports

A.1 General

SQL language reference for AdvaInform Reports Relational Database gives a description of SQL commands which are allowed to use towards the Relational Database. Interactive access towards the Relational Database is performed via the Template Builder.

A.2 SELECT

A.2.1 Purpose

Display rows and columns from one or more tables. May be used as a statement, or (with certain restrictions on clauses) as a query or subquery in another statement.

A.2.2 Prerequisites

Must own the table, or have SELECT privilege on the table(s), or have DBA privilege.

A.2.3 Syntax

```
SELECT [ALL|DISTINCT] { * I table.* I expr [c alias] }  
      FROM [user.]table [t alias] [, [user.]table [t_ alias]]...  
      [WHERE condition]  
      [GROUP BY expr[, expr]... [HAVING condition]]  
      [{UNION|INTERSECT|MINUS} SELECT... j  
      [ORDER BY {expr|position} [ASC|DESC]  
      [, [expr|position} [ASC|DESC]]]...
```

A.2.4 Keywords and Parameters

ALL: is the default and specifies that all rows that satisfy the SELECT statement should be displayed.

DISTINCT: specifies that duplicate rows should be removed before the rows are returned. A row is considered a duplicate of another if every value for each column of the SELECT clause matches that of another row(s).

*: specifies that all columns from all tables listed in the FROM clause to be displayed.

table.*: specifies that all columns from the named table to be displayed.

c_alias provides a different name for the column expression and causes the alias to be used in the column heading. Does not affect the actual name of the column. Column aliases cannot be referenced elsewhere in a query.

[user.]table	specifies which table(s) or view(s) to display, optionally prefaced by the owner name of the table. If not specified, the owner name defaults to the current user.
t_alias	provides a different name for the table for the purpose of evaluating the query and is most often used in a correlated query. Other references to the table throughout the query must refer to the alias name.
positions	identifies a column based on its relative position in the SELECT column list rather than by its name.
ASC DESC	specifies the column should be sorted ascending or descending based on the operating system collating sequence.

Many of the SELECT clauses are individually described in [Section A.3, SELECT STATEMENT CLAUSES](#).

A.2.5 Usage Notes

Clauses must be used in the order shown above except that:

- GROUP BY and HAVING may be specified in any order.
- ORDER BY may be interchanged.

Each expr becomes the name of one column in the display, and each table.* becomes a set of columns, one for each column in table in the order they were defined when the table was created. The width and datatype of each expression is determined by the elements of the expression.

The asterisk character, *, represents all columns in a table. If multiple tables are specified in the FROM clause, then * represents all columns from all tables. The asterisk may be prefaced by table., which limits the columns to those belonging to the named table.

Each alias, if specified, is used to label the preceding expr in the table so that the column is displayed with a new heading. An alias has no effect on the actual name of the column, and cannot be referenced in the remainder of the query.

FROM table specifies the table or view to be selected from. More than one table implies a join. A table alias value, if specified, must be used as an alias for the table throughout the entire SELECT statement.

WHERE specifies a condition to be used to select rows and /or join tables. This condition may often contain subqueries a nested SELECT statement.

A.2.6 GROUP BY, HAVING

The GROUP BY and HAVING clauses are used to display summary information about groups of rows that have the same values in one or more fields. Refer to the GROUP BY clause in [Section A.3, SELECT STATEMENT CLAUSES](#).

A.2.7 Set Operators

The UNION, INTERSECT, and MINUS operators combine the results of two SELECT statements into a single result. UNION returns those results returned by either component query and eliminates duplicate results; INTERSECT returns only those rows returned by both queries; and MINUS returns those rows returned by the first SELECT after removing those rows that also are returned by the second SELECT. The number and types of the columns selected by each component query must be the same (but the lengths need not be the same).

If more than two SELECTs are combined in this way, adjacent pairs of SELECTs are evaluated from the top down. (Parentheses may be used to specify evaluation in a different order.) All three operators imply DISTINCT in all SELECTs.

A.2.8 ORDER BY

The ORDER BY clause specifies the order in which rows of the table are returned. Refer to the ORDER BY clause in [Section A.3, SELECT STATEMENT CLAUSES](#).

A.2.9 Sequences

A SELECT statement may also refer to the sequence pseudo-columns NEXTVAL and CURRVAL.

A.2.10 Subquery

Refer to the syntax description of query in [Section A.3, SELECT STATEMENT CLAUSES](#).

A.2.11 Examples

To display ENAME, SAL, JOB, and DEPTNO from all rows of the table EMP in which the department number is 30 or less, enter:

```
SELECT ENAME, SAL, JOB, DEPTNO
FROM EMP
WHERE DEPTNO=30
```

A.2.12 Related Topics

The FROM, WHERE, GROUP BY, HAVING, ORDER BY clauses; the syntax descriptions of simple join, outer join, and query in, [Section A.3, SELECT STATEMENT CLAUSES](#).

A.3 SELECT STATEMENT CLAUSES

In this section, each clause of the SELECT statement is described in detail. Additional uses of the SELECT statement are also covered.

Topics include:

- FROM clause
- WHERE clause
- GROUP BY and HAVING clauses
- ORDER BY clause
- simple joins
- outer joins
- query
- correlated queries
- distributed queries.

A.3.1 FROM clause

A.3.1.1 Purpose

Specifies a table or tables for a SELECT statement.

A.3.1.2 Prerequisites

None.

A.3.1.3 Syntax

In SELECT statements:

```
SELECT select list  
FROM [user.] table [alias] [@databaselink] [...]  
WHERE...  
...
```

A.3.1.4 Keywords and parameters

user: is the owner of the table or view.

table: is the name of the table or view.

alias: a temporary name assigned to a table or column an alias allows the same table to be referenced in a correlated query. The alias also allows you to abbreviate a table name within the query.

A.3.1.5 Usage Notes

FROM is a required clause in the SELECT statement. It identifies the table(s) from which rows are selected.

If the table is owned by another user, the table name must be qualified with the owner's username. The username and table name are separated by a period.

A.3.1.6 Examples

To select rows with the value 40 in the DEPTNO column from a table named EMP, enter:

```
SELECT * FROM EMP  
WHERE DEPTNO = 40
```

To select rows with the value 40 in the DEPTNO column from two tables named EMP and EMP2, enter:

```
SELECT *  
FROM EMP, EMP2  
WHERE EMP.EMPNO = EMP2.EMPNO  
AND DEPTNO = 40
```

If EMP2 is owned by another user whose user ID is BEN, enter:

```
SELECT *  
FROM EMP, BEN.EMP2  
WHERE EMP.EMPNO = EMP2.EMPNO  
AND DEPTNO = 40
```

A.3.1.7 Related Topics

SELECT statements in this appendix.

A.3.2 GROUP BY and HAVING clauses

A.3.2.1 Purpose

Operate on groups of rows that are returned in the query and limit (with the HAVING clause) which groups are affected.

A.3.2.2 Prerequisites

None.

A.3.2.3 Syntax

```
SELECT expr [,expr]
      FROM table [,table].
      GROUP BY expr [,expr]
      HAVING condition
      ...
```

A.3.2.4 Keywords and Parameters

table: is a table or view to which you have access.

A.3.2.5 Usage Notes

Use the GROUP BY clause to make the SELECT statement retrieve one summary row for each group of selected rows. Each group of selected rows will be formed on the basis of having the same value for the column(s) or expression(s) of the GROUP BY clause.

Each SELECT expr must either:

- be a constant or a function with no parameters (e.g. SYSDATE).
- contain a grouping function such as SUM, COUNT, or MAX
- exactly match an expr in the GROUP BY clause

GROUP BY expressions can reference any column in the tables in the FROM clause, whether or not the columns are mentioned in the SELECT list.

Specify GROUP BY and HAVING after WHERE. If both the GROUP BY and HAVING clauses are specified, they may be specified in either order.

There is a system-dependent limit on the total number of bytes in all expressions in the GROUP BY clause.

A.3.2.6 HAVING clause

Use the HAVING clause to specify which GROUP BY groups are to be included in the result. ORACLE processes the WHERE, GROUP BY, and HAVING clauses in the following manner:

1. ORACLE removes all rows that do not satisfy the WHERE clause.
2. ORACLE calculates and forms the groups as specified in the GROUP BY clause.
3. ORACLE removes all groups that do not satisfy the HAVING clause.

A.3.2.7 Examples

To display the minimum and maximum salary for each department in the table EMP, enter:

```
SELECT DEPTNO, MIN(SAL), MAX(SAL)
FROM EMP
GROUP BY DEPTNO
```

Figure A-1. Data for query

DEPTNO	MIN(SAL)	MAX(SAL)
10	1300	5000
20	800	3000
30	950	2850

To apply this query only to clerks, enter:

```
SELECT DEPTNO, MIN(SAL), MAX(SAL)
FROM EMP
WHERE JOB = 'CLERK'
GROUP BY DEPTNO
```

Table A-1. Data for query

DEPTNO	MIN(SAL)	MAX(SAL)
10	1300	1300
20	800	1100
30	950	950

To make the query display only departments whose lowest salary is below \$1,000:

```
SELECT DEPTNO, MIN(SAL), MAX(SAL)
FROM EMP
WHERE JOB = 'CLERK'
GROUP BY DEPTNO
```

HAVING MIN(SAL) < 1000

Table A-2. Data for query

DEPTNO	MIN(SAL)	MAX(SAL)
20	800	1100
30	950	950

A.3.3 WHERE clause

A.3.3.1 Purpose

Restrict which rows are returned from the table in a SELECT statement.

A.3.3.2 Prerequisites

None.

A.3.3.3 Syntax

SELECT...

WHERE condition

A.3.3.4 Keywords and Parameters

None

A.3.3.5 Usage Notes

The purpose of a WHERE clause is to specify a condition. The statement containing the WHERE clause will act only on rows that satisfy the condition.

The condition is defined by a logical expression. For example,

DEPTNO=30

is a logical expression that is true if the value of DEPTNO is 30, and otherwise is false.

AND and OR may be used to form compound logical expressions. For example, the expression:

DEPTNO=30 AND JOB=' SALESMAN'

is true only if both of the component expressions are true.

NOT may be used to reverse the sense of an expression. For example,

NOT (DEPTNO=30 AND JOB=' SALESMAN'

is true if:

DEPTNO=30 AND JOB=' SALESMAN'

is false.

A.3.3.6 Examples

This statement selects all rows in which the value of DEPTNO is not equal to 30:

```
SELECT *
FROM EMP
WHERE DEPTNO!=30
```

A.3.3.7 Using the TO_DATE Function in a Condition

The TO_DATE function has the following syntax:

```
TO_DATE(char[,fmt])
```

The purpose is to convert *char* to a date value. The current month, year, etc. are used to provide components of the date not explicitly determined from the arguments. *fmt* is a char value in date format specifying the format of *char*. If *fmt* is omitted, *char* must have the default format, "DD-MON-YY". If *fmt* is J, for Julian, then *char* should actually be a number.

Date formats

Date formats are used in the TO_DATE function (as shown above) to supply values to Oracle in a nonstandard date format. As mentioned above the standard date format is DD-MON-YY. The following table lists the elements of a date format model.

Table A-3. Date formats in the TO_DATE function

Element	Meaning
SCC or CC	Century;"S" prefixes "BC" date with "-"
YYYY or SYYY	4-digit year; "S" prefixes "BC" date with "-"
YYY or YY or Y	Last 3,2, or 1 digit(s) of year
Y,YYY	Year with comma in this position
SYEAR or YEAR	Year, spelled out; "S" prefixes "BC" date with "-"
BC or AD	BC/AD indicator
B.C. or A.D.	BC/AD indicator with periods
Q	Quarter of year (1,2,3,4; JAN-MAR=1)
MM	Month (01-12; JAN=01)
MONTH	name of month, padded with blanks to length of 9 characters
MON	Name of month, 3-letter abbreviation

Table A-3. Date formats in the TO_DATE function

Element	Meaning
WW	Week of year (1-52) where week 1 starts on the first day of the year and continues to the seventh day of the year
W	Week of month (1-5) where week 1 starts on the first of the month and continues to the seventh of the month
DDD	Day of year (1-366)
DD	Day of month (1-31)
D	Day of week (1-7)
DAY	Name of day, padded with blanks to length of 9 characters
DY	name of day, 3-letter abbreviation
J	Julian day; the number of days since January 1, 4712 BC
AM or PM	Meridian indicator
AM or PM	Meridian indicator with periods
HH or HH12	Hour of day (1-12)
HH24	Hour of day (1-23)
MI	Minute (0-59)
SS	Second (0-59)
SSSSS	Seconds past midnight (0-86399)
/,.,	Punctuation is reproduced in the result
...	Quoted string is reproduced in the result

Prefix *FM* - "Fill Mode":

This prefix suppresses blank padding for all subsequent character elements (such as MONTH) and suppresses leading zeroes for all subsequent number elements (such as MI) in a date format model. FM is a toggle. If FM appears twice in a date format model, blank padding and leading zeroes are suppressed for elements after the first FM and reenable for elements after the second.

Capitalization in a spelled-out word or abbreviation follows capitalization in the corresponding format element. For example, "DAY" in a format model produces capitalized words like "MONDAY"; "Day" produces "Monday"; and "day" produces "monday".

A date format model may also include punctuation characters like hyphens, slashes, and commas, and character constants enclosed in quotation marks (not apostrophes). Such punctuation characters and character constants will be reproduced in the output.

For example, the format model:

```
'DDTH "of" fmMonth, YYYY'
```

produces a date date like:

```
15TH of March, 1989
```

Note that the fill mode (*fm*) format mask was also used. If omitted, the month would be blank padded to 9 characters, as in:

```
15TH of March      , 1989
```

To include an apostrophe in a date format element, use two apostrophes in a row. For example:

```
SELECT TO_CHAR (sysdate, 'fmDay'''s Special''')  
FROM DUAL
```

would yield:

```
Tuesday's Special
```

A.3.4 ORDER BY Clause

A.3.4.1 Purpose

Sort the result of a query.

A.3.4.2 Prerequisites

None.

A.3.4.3 Syntax

```
SELECT...
```

```
ORDER BY {expr|position} [ASC|DESC],.
```

A.3.4.4 Keywords and Parameters

expr: is an expression based on one or more of the columns in the SELECT clause or from one of the tables in the FROM clause. For a closer description of *expr*, see [Section A.3.4.5, Expr\(ession\)](#).

position: is a number identifying the position of a column in the SELECT clause. For example, rather than ordering by column names, position allows you to order by the first column in the select list. If the UNION, INTERSECT, or MINUS operator is used, ORDER BY items must be specified by position.

ASC | DESC: specifies ascending or descending order according to the collating sequence for the character set. Null values follow all others in ascending order and precede all others in descending order. Thus, nulls sort “high”.

A.3.4.5 Expr(ession)

An *expression* is a combination of one or more values, operators, and functions which evaluates to a value. An expression generally assumes the datatype of its components. For a formal definition of an Oracle expression, see below.

This is an example of a simple expression that evaluates to 4 and has datatype NUMBER (the same datatype its components):

```
2*2
```

This is an example of a more complex expression that uses both functions and operators. The expression adds seven days to the current date, removes the time component from the sum, and converts the result to CHAR datatype:

```
TO_CHAR( TRUNC( SYSDATE+7 ) )
```

You can use an expression almost anywhere you can use a constant value.

A.3.4.6 Usage Notes

Use the ORDER BY clause to specify the order in which the results of a query are to be displayed. The ORDER BY clause is the only method to guarantee the ordering of rows.

Rows are ordered by the value of the first expr or position. Rows with the same value are ordered by the value of the second item (if one is specified), and so on.

Sorting by position is useful when sorting by a long expression. Rather than retyping the entire expression in the ORDER BY clause, you can just supply the expression's position in the SELECT clause. Sorting by position is necessary when using UNION, INTERSECT, or MINUS because the tables referenced by these operators need not have columns with the same name.

You may not reference a column alias in the ORDER BY clause.

The ORDER BY clause is subject to the following restrictions:

- If the ORDER BY clause and the DISTINCT operator are both specified, the ORDER BY clause may not refer to columns that are not mentioned in the SELECT clause.
- A system-dependent limit applies to the total number of bytes in all expressions in the ORDER BY clause.

A.3.4.7 Example

To select all salesmen's records from EMP, and order the output by commission in descending order, enter:

```
SELECT *  
FROM EMP
```

```
WHERE JOB = 'SALESMAN'
```

```
ORDER BY COMM DESC
```

To select the employees from EMP ordered first by ascending department number and then by descending salary, enter:

```
SELECT ENAME, DEPTNO, SAL
```

```
FROM EMP
```

```
ORDER BY DEPTNO ASC, SALARY DESC
```

To select the same information as the previous SELECT and use the positional ORDER BY notation, enter:

```
SELECT ENAME, DEPTNO, SAL
```

```
FROM EMP
```

```
ORDER BY 2 ASC, 3 DESC
```

A.3.4.8 Related Topic

SELECT statement in this appendix.

A.3.5 Simple Join

A.3.5.1 Purpose

Combine data from multiple tables. Result may be displayed or may be used as input for another SQL statement.

A.3.5.2 Prerequisites

Must have SELECT access to or must be the creator of each table referenced.

A.3.5.3 Syntax

```
SELECT [{table|table alias}.] column
```

```
[,[{table|table_alias}.]column]...
```

```
FROM table [table_alias], table [table_alias]...
```

```
[WHERE {table|table_alias}.column = {table|table_alias}.column]
```

A.3.5.4 Usage Notes

A join is the form of the SELECT statement that combines rows from two or more tables. Thus, each row of the result will have column data from more than one table. A join occurs whenever multiple tables are referenced in the FROM clause of the SELECT statement. The optional WHERE clause determines how the rows of the tables are combined. The following discussion assumes a join of two tables, but may be applied to joins of two or more tables.

The most common type of join, the simple join, returns rows from two tables based on an equality condition. This condition appears in the WHERE clause in the form of:

`<table1.columnname> = <table2.columnname>`

The rows of table1 are combined with the rows of table2 and the resulting rows are returned where the values of table1.columnname and table2.columnname are equal.

The columns used in the WHERE condition need not have the same name. If the column names are identical, you must preface the columns with the table name so there is no ambiguity.

If a table alias is used in place of the actual table name in the FROM clause, then that table alias must be substituted for the table name throughout the SELECT statement including references to the table in the SELECT list and WHERE clause.

ORACLE places no restrictions on which columns from either table are actually displayed; none, some, or all columns from both tables may be returned. Thus, the columns specified in the WHERE clause need not appear in the SELECT clause.

A simple join is also called an equi_join because it uses an = comparison operator in the WHERE clause.

A.3.5.5 Cartesian product

A cartesian product is formed when tables are joined without the use of a WHERE clause. The omission of the WHERE clause causes all rows from all tables to be combined. A cartesian product always generates a large number of rows and its result is rarely useful. For example, if two tables each have one hundred rows, the resulting cartesian product will be ten thousand rows.

You should always include a WHERE clause when joining tables unless you have a specific need to combine all rows of all tables.

A.3.5.6 Examples

The following simple join answers the question, "What are the name and number of the department in which each employee works":

```
SELECT ENAME, EMP.DEPTNO, DNAME
FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
```

Table A-4. Data for query

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
MILLER	10	ACCOUNTING
KING	10	ACCOUNTING

Table A-4. Data for query

ENAME	DEPTNO	DNAME
SMITH	20	RESEARCH
SCOTT	20	RESEARCH

A join is required because the data is stored in two different tables. Rows of the two tables are joined according to the WHERE condition:

EMP.DEPTNO = DEPT.DEPTNO.

A.3.5.7 Related Topics

The syntax description of outer join later in this section; SELECT statement in this appendix.

A.3.6 Outer Join

A.3.6.1 Purpose

Join two or more tables together and also return those rows from one table which have no direct match in the other table.

A.3.6.2 Prerequisites

Must have SELECT privileges on or own each table referenced.

A.3.6.3 Syntax

```
SELECT  
  
FROM table1, table2...  
  
WHERE table1.column = table2.column (+i)
```

A.3.6.4 Usage Notes

The outer join extends the result of a normal join. An outer join returns all the rows returned by the simple join and it also returns those rows from one table that do not match any row from the other table. For example, if we are selecting all employees from EMP and their locations from DEPT, there may be an employee whose dept is not listed in DEPT. We'd like to see his record along with the other employee's.

An outer join must contain a WHERE clause with a condition in one of these forms:

```
table1.column = table2.column (+)  
table1.column (+) = table2.column
```

Extra “null” columns will be “created” temporarily for the table with the (+) outer join operator and joined against all rows from the other table that would have not been returned in a normal join.

The outer join symbol must follow a column reference within a join condition.

Within a single SELECT statement, a table may be outer joined to at most one other table.

A.3.6.5 Example

To display the total monthly salary for all departments (without an outer join), enter:

```
SELECT DEPT.DEPTNO, DNAME, SUM(SAL) MONTHLY SALARY
FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
GROUP BY DEPT.DEPTNO, DNAME
ORDER BY DEPT.DEPTNO
```

To display the total monthly salary for all departments, even those

Table A-5. Data for query

DEPTNO	DNAME	MONTHLY SALARY
10	ACCOUNTING	8750
20	RESEARCH	10875
30	SALES	9400

To display the total monthly salary for all departments, even those departments with no employees (with an outer join), enter:

```
SELECT DEPT.DEPTNO, DNAME, SUM(SAL) MONTHLY SALARY
FROM EMP, DEPT
WHERE EMP.DEPTNO(+) = DEPT.DEPTNO
GROUP BY DEPT.DEPTNO, DNAME
ORDER BY DEPT.DEPTNO
```

Table A-6. Data for query

DEPTNO	DNAME	MONTHLY SALARY
10	ACCOUNTING	8750
20	RESEARCH	10875

Table A-6. Data for query

DEPTNO	DNAME	MONTHLY SALARY
30	SALES	9400
40	OPERATIONS	

In the preceding outer join, the OPERATIONS department was displayed even though no employees in the EMP table belonged to the OPERATIONS department.

The rule of thumb is to append the outer join symbol following the name of the table without the matching rows.

A.3.6.6 Related Topics

The syntax description of simple join in [Section A.3.5, Simple Join](#); SELECT statement in this appendix.

INDEX

A

Action Menu 4-20, 4-22
Activate 4-3
AdvaInform
 History 3-1
 Reports 1-1
 SQL*Connect 1-8, 3-2
 User API 1-8, 3-1
AdvaInform SQL*Connect 3-101
Application Programmers Interface 1-8

B

Basic Object
 reading 3-101
Batch 3-2
Build Templates Menu Item 3-83

C

CO_xx_INST table 3-101
Condition List 1-19
 Modify 3-65

D

Deactivate 4-3
Delete 3-83
diagnostic messages 5-2

E

Encapsulated Data 1-8
Encapsulation 3-41
Error Messages 5-2
error messages 5-13
Event Conditions 3-80
Execution 4-6
Execution Parameters 1-8, 1-16, 1-18, 3-45, 3-65
 File 1-18, 3-65
 History 1-18, 3-65
 Printer 1-18, 3-65
Exit 3-83

F

File 3-67
File Menu 3-82
Finished Report 1-9
FROM 3-85

G

Group 3-63

H

Historical Storage 4-9
History 3-67

L

location table 3-101
Logs 3-2

M

Manual 3-64
Manual Execution 1-21, 3-83, 4-6
Manual Report Execution Window Menu 4-19
messages 5-2

N

NUMLOGVAL 3-90
 ENTRYSTATUS 3-91
 ENTRYVALUE 3-91
 Name 3-91
 OBJECTSTATUS 3-91
 Time 3-91

O

object types
 IMS Basic Objects 3-101
 IMS User Objects 3-101
Objects 1-11
Open 3-82
Operator Functions 4-1
ORDER BY 3-85

P

Parameter 1-18, 3-65
Parameter box 3-68
Parameter List 1-18
 Delete 3-64
 Modify 3-64
Printer 3-67
Progra 3-64

R

RDB 1-8
read

R (continued)

- Basic objects 3-101
- User Objects 3-101
- Refresh 4-20
- Relational Database 1-8
- Report
 - Archive 4-16
 - Configuration 1-1, 1-14
 - Delete 4-14
 - Execution 1-1, 1-9
 - Format 1-16
 - Log 1-9, 3-2
 - Modify 4-13
 - Object Instance 1-8, 1-17
 - Presentation 1-9
 - Print 4-12
 - Search 4-16
 - Status 1-1, 1-22, 4-2
 - Template 1-14, 3-39, 3-64
 - View 4-9
- Report Configuration File Menu 3-82
- Report Configuration Window 1-14, 3-63
- Report Configuration Window Menu 3-83
- Report Execution 1-24
- Report Status 3-83
- Report Template 1-8
- Reports List Action Menu 4-22
- Reports List Window Menu 4-22
- Reports Menu 3-62

S

- Save 3-83
- Save As 3-83
- Schedule Mode 1-18
- Scheduling 4-3
 - Activate 4-3
 - Day Configuration 3-72
 - Deactivate 4-3
- Scheduling Modes 4-3
 - Manual 4-5
 - Program 4-5
 - System 4-5
- SELECT 3-85, A-1
- Select 1-13
- Select Statement 1-13
- SQL 1-8, 1-14, 3-39
- SQL*Connect 3-101
- SQL*ReportWriter 1-14, 3-39
- Start-up 2-1

S(continued)

- Structured Query Language 1-8
- System 3-64

T

- TCL 3-2
- Template 3-64

U

- User Objects
 - reading 3-101

V

- View Logs 3-83
- View Logs Action Menu 4-20
- View Logs File Menu 4-20
- View Logs Window Menu 4-21

W

- WHERE 3-85
- Wildcard Characters 3-100
- Window Menu 3-83, 4-19, 4-21 to 4-22



3BUR 001 405 R0001

September 1994